Linux Network Servers 452

Ver-11-2009

www.4linux.com.br

Sumário

Índice de tabelas

Índice de Figuras

Capítulo 1

Ajustes Iniciais

1.1. Objetivos

- Configurar a rede;
- Configurar resolução de nomes;
- Configurar hostname;
- Desativar Serviços desnecessários;
- Configurar repositório de rede.

1.2. Introdução teórica

Antes de iniciar o trabalho de instalação e configuração de serviços, é necessário checar se algumas configurações são válidas. Utilize o comando:

```
#  si_cliente
```

Baixar uma imagem padrão com Debian Lenny e Gnome instalado direto do servidor, por favor não troque a senha padrão 123456 que utilizamos, para que o pessoal do suporte possa sempre ajudar em caso de problemas.

Para saber mais sobre o System Imager, que é o sistema de salvamento de imagens, acesse:

```
http://wiki.systemimager.org
```

1.3. Configurando a rede

Vamos nos certificar que o arquivo /etc/network/interfaces está corretamente configurado:

 # vim /etc/network/interfaces

```
1 # Interface Loopback
2 auto lo
3 iface lo inet loopback
4
5 # Interface de rede primária
6 auto eth0
7 iface eth0 inet static
8        address192.168.200.X
9        netmask255.255.255.0
10        network192.168.200.0
11        broadcast 192.168.200.255
12        gateway192.168.200.254
```

Red Hat:

 # vim /etc/sysconfig/network/ifcfg-eth0

1.4. Configurando a resolução de nomes

1) Edite o arquivo /etc/resolv.conf:

```
# vim
```

2) Adicione o endereço IP do servidor DNS da Embratel:

```
nameserver 200.176.2.10
```

1.5. Configuração do hostname

Em um servidor, a configuração correta do hostname e do arquivo hosts é essencial para eliminar problemas em serviços de rede. Abra o arquivo /etc/hostname, adicione o nome, x é o número de sua máquina (caso esteja errado):

```
# vim /etc/hostname
 microX
```

1) Edite o arquivo de /etc/hosts:

```
# vim /etc/hosts
```

```
127.0.0.1 localhost.localdomain localhost
 192.168.200.X microX.microx.com.br microX
```

1.6. Configurando o repositório de rede

Para instalar programas no Debian GNU/Linux, é necessário que o repositório de rede corretamente configurado. Vamos configurá-lo para buscar os softwares em um repositório interno da 4linux.

1) Abra o arquivo responsável pela configuração do repositório:

```
# vim   /etc/apt/sources.list
```

2) Edite-o apontando para o repositório interno da 4Linux:

```
deb http://192.168.1.1/debian  lenny  main  contrib  non-free
```

3) Depois de configurado, faça o update do banco de dados do aptitude:

```
# aptitude update
```

4) Agora, vamos instalar e configurar o nosso editor de textos habilitando destaque de sintaxe e linhas numeradas:

```
# aptitude install vim
# echo "syntax on" >> /etc/vim/vimrc
# echo "set number" >> /etc/vim/vimrc
```

Procedimentos de Pós Instalação, conhecidos também como Hardening, são assuntos discutidos no treinamento de Segurança 415 – ISO27002.

1.7. Remover serviços desnecessários

Todo Administrador de Sistemas tem, ou deveria ter ciência de que um servidor deve executar somente os softwares necessários, eliminando assim riscos de segurança e ajudando a melhorar a performance.

Para que isso seja feito, precisamos antes verificar quais serviços de rede estão habilitados e aceitando conexões.

1) Verificando serviços de rede com netstat:

```
# netstat -nltup
```

2) Encerrando os serviços encontrados:

```
# invoke-rc.d exim4 stop
# invoke-rc.d nfs-common stop
# invoke-rc.d portmap stop
# invoke-rc.d openbsd-inetd stop
```

3) Removendo serviços da inicialização:

```
# update-rc.d -f exim4 remove
# update-rc.d -f nfs-common remove
# update-rc.d -f portmap remove
# update-rc.d -f openbsd-inetd remove
```

Red Hat:

 # chkconfig --list

 # chkconfig --level 2345 portmap stop

O comando runlevel, mostra qual é o nível de inicialização que nos encontramos.

Você também pode habilitar os serviços, com o programas sysv-rc-conf e com o rcconf, que foram utilizados no treinamamento 451:

aptitude install rcconf sysv-rc-conf

rcconf

#sysv-rc-conf

1.8. Definindo variáveis e alias de sistemas

O Sistema Operacional vem habilitado por padrão sem alias e somente as variavéis essenciais ao seu funcionamento. Então iremos configurar nosso GNU/Linux.

1) Agora vamos colocar as variáveis e alias, para isso, edite o arquivo:

```
# vim /etc/profile
```

2) Vamos inserir estas opções no final do arquivo:

```
alias ls='ls –color'
alias lsl='ls -l'
TMOUT=1200
HISTSIZE=1000
```

3) Para validar as mudanças no disco, execute:

```
# source /etc/profile
```

Os comandos set e env, vão ser cobrados na prova do LPI, eles são uteis para listarem as variavéis globais e locais.

1.9. Funcionamento do Sistema

Todos os itens que veremos agora já conhecemos, mas são cobrados na LPI.

1) Para identificar a versão do kernel instalado e distribuição:

```
# uname -r
# cat   /pro/version
```

2) O kernel fica carregado na memória RAM, este é seu arquivo:

```
# cd  /boot
# file  vmlinuz-2.6.26-2-686
# du  -sh  vmlinuz-2.6.26-2-686
```

3) Logo que o kernel é carregado em memória, logo inicia-se os processos:

```
# cd  /proc
# ls
# ps  aux
# cat  cpuinfo
```

4) Os arquivos do diretório /etc, são lidos, onde são montados os dispositivos e a inicialização dos serviços:

```
# cat   /etc/fstab
# cat   /etc/mtab
```

5) Repare o que está montado no sistema:

```
# cat   /proc/mounts
# cat   /proc/partitions
# df  -h
```

6) Os dipositivos ficam /dev, e seu sistema de arquivos é o udev.

```
# ls  -l  /dev
```

Veja novamente estes tópicos no 450 e no 451.

7) Os módulos carregados no sistema:

```
# ls  -l  /lib/modules
# lsmod
# modprobe -l
# cat  /proc/modules
```

8) Para exibir todas bibliotecas do sistema:

```
# ldconfig  -p
# ls /lib
# cat  /etc/ld.so.conf
```

Fique atento, porque tudo que estudamos no primeiro capítulo, irão cair no exame da LPI, estude as apostilas da formação. Para informações acesse: http://www.lpibrasil.com.br/

1.10. Exercícios Teóricos

1) Qual a importância dos procedimentos de pós-instalação?

2) Qual é o diretório menos importante fazer backup no Ssitema? (FHS)

3) Como eu consigo verificar quais arquivos tem permissões "especiais" habilitadas de SGUID e SUID?

4) Qual a máscara padrão do sistemas para criação de arquivos e diretórios, qual arquivo ela fica armazenada?

5) Como ficaria as permissões de arquivos e diretórios, se a umask do sistema fosse 23 e 65?

6) Quais diretórios tem stick bit ativado?

1.11. Laboratório

1. Instale o htop para gerênciar os processos;
2. Desabilite o gdm do runlevel padrão, utilizando sysv-rc-rconf.

Capítulo 2

PAM

2.1. Objetivos

- Entender a configuração do PAM;
- Conhecer seus módulos;
- Entender seus controles;
- Impedir login de super usuário via console.

2.2. Introdução teórica

A cada dia, mais e mais mecanismos de autenticação diferentes surgem no mercado. Imagine que para cada método de autenticação, fosse necessário rescrever aplicações como FTP, apache, ssh e o sistema de login console e gráfico do Linux?

Foi pensando neste tipo de esforço que foi criado o mecanismo de autenticação PAM ou Pluggable Authentication Modules, que oferece uma camada de abstração para autenticação em sistemas Unix. Assim, se quisermos adicionar um novo modelo de autenticação via leitor de digitais, por exemplo, basta apenas instalar o módulo PAM que ofereça essas funcionalidades e configurar as aplicações para trabalhar com este módulo, sem a necessidade de recriar ou reimplementar funções no aplicativo.

Além disso, com o PAM é possível controlar horários e terminais disponíveis para logins, quais serão os usuários que podem efetuar um su no sistema operacional, autenticar em bases de dados diferentes, além de outros módulos que

estão disponíveis no seguinte endereço:

```
http://www.kernel.org/pub/linux/libs/pam
```

2.3. Módulos

O PAM trabalha com módulos e controles, e cada tipo de módulo provem uma funcionalidade diferente dentro do sistema. Vamos comentar primeiro os módulos:

- **account:** Verifica se a conta é valida no sistema, se a password do usuário expirou e se o usuário realmente tem direitos de acessar aquele serviço.
- **authentication:** Verifica questões de autenticação, seja por senhas ou impressões digitais (quando falamos de biometria). É o módulo authentication que oferece a flexibilidade do PAM, já que, para cada método de autenticação criado, existe uma biblioteca que será adicionada à este módulo.
- **password:** Este módulo é responsável por cuidar dos aspectos relacionados a tarefas envolvendo senhas, como atualização e solicitação de nova senha de acesso no caso da troca da mesma.
- **session:** Responsável por tarefas pós autenticação, como montar um compartilhamento de arquivos remotos que contém o diretório /home do usuário em questão, por exemplo.

2.4. Controles

Além dos módulos, existem também os controles, os quais comentaremos agora:

- **required:** Checa a existência do módulo solicitado, caso esse módulo falhe, somente depois de verificar todos os módulos do mesmo tipo disponíveis é que o usuário será avisado.
- **requisite:** Checa a existência do módulo solicitado e avisa o usuário imediatamente caso este módulo falhe.

- **sufficient:** Somente a verificação do módulo é suficiente para a autenticação, desde que nenhum módulo marcado como required falhe.
- **optional:** O sucesso ou a falha deste módulo não interfere no processo de autenticação.

> *A maioria das distribuições trabalham com o PAM, as únicas distribuições que não trabalham com PAM são as baseadas em* ***Slackware.***

2.5. Prática dirigida

1) Vamos verificar quais módulos do PAM já estão instalados em nosso sistema:

```
# ls /lib/security/
```

2) Vamos olhar dentro do diretório do PAM procurando por arquivos de configuração para os programas instalados:

```
# ls /etc/pam.d
```

3) Para entendermos como o PAM funciona, vamos ativar um módulo simples que serve para bloquear usuários comuns. Edite o arquivo /etc/pam.d/login e visualize o módulo pam_nologin.so:

```
# vim /etc/pam.d/login
```

```
auth requisite pam_nologin.so
```

4) Para esse plugin funcionar ele necessita que o arquivo nologin esteja criado dentro do diretório /etc. Crie o arquivo nologin dentro do diretório /etc:

```
# touch /etc/nologin
```

Com isso tente logar com um usuário comum em outro terminal

Repare que a partir do momento que o plugin está ativado no programa login e o arquivo necessário está criado, os usuários comuns não conseguem mais logar, somente o usuário root. Isso não é muito viável mas serve de exemplo para entendermos como o PAM funciona.

Uma maneira prática de usar o PAM é fazermos com que o usuário root não tenha acesso direto ao login, forçando a logar com usuário comum e depois fazer um su para virar root.

5) Para isso aconter, editamos o arquivo e visualizamos a regra:

```
# vim /etc/pam.d/login
```

```
account requisite pam_time.so
```

6) Edite o arquivo time.conf dentro de /etc/security e acrescente na última linha:

```
# vim /etc/security/time.conf
```

```
login;*;root;!Al0000-2359
```

Os campos acima são:

- login - Serviço que irá ser controlado
- * - Terminal
- root - Usuário
- Al0000-2359 - Dias e horários de filtragem.

Efetue login com o usuário root em outro terminal e veja que não será possível efetuar o login.

Com o PAM, podemos limitar quais usuários poderão ter acesso a utilizar o comando su. Para isso, crie um grupo chamado admins para os usuários que poderão ter acesso a fazer o su.

7) Adicione o grupo onde os usuários que poderam fazer o su:

```
# groupadd admins
```

8) Agora vamos adicionar o usuário aluno ao grupo admins:

```
# adduser aluno admins
```

9) Crie uma política que não possibilite o uso de su, exceto pelos usuários do grupo admins:

```
# vim /etc/pam.d/su
```

```
auth requiredpam_wheel.so group=admins
```

Para fazer o teste, logue-se como o usuário que pertence ao grupo admins e tente virar root usando o su.

10) Em seguida tente com um usuário que não pertence ao grupo admins:

```
$ su -
```

11) No final dos arquivos possibilite ssh no horário das 7:30 às 19:00. Para isso insira o módulo pam_time.so no arquivo do ssh em /etc/pam.d:

 # aptitude install ssh

```
# vim /etc/pam.d/ssh
account  required  pam_time.so
```

12) Em seguida coloque a regra no time.conf:

```
# vim /etc/security/time.conf
```

```
sshd;*;*;Al0730-1900
```

2.6. Exercícios Teóricos

1) Qual é a função do PAM e como ele trabalha?

____

____

____

2) Quais aplicações podem ser integradas com o PAM?

____

____

3) Qual a diferença do required para o requisite nas diretivas do PAM?

____

____

2.7. Laboratório

1. Permita o login de root apenas no tty6;
2. Crie o usuário "manutencao" faça com ele efetua login ssh apenas de segunda a sexta feira, das 8:00 as 18:00.

Capítulo 3

RAID

3.1. Objetivos

- Entender os principais níveis de RAID;
- Configurar RAID-1;
- Verificar o estado do RAID;
- Simular falhas no RAID.

3.2. Introdução teórica

O RAID (Redundant Array of Inexpensive Disks) foi desenvolvido em 1988 como uma solução barata para garantir a disponibilidade da informação armazenada em discos, utilizando para isso uma configuração especial de discos rígidos, que podem oferecer redundância em caso de falhas e ganho de performance em escrita ou leitura, dependendo da configuração do conjunto RAID.

Como principais vantagens, o RAID oferece:

- Ganho de desempenho no acesso para leitura ou gravação;
- Redundância em caso de falha em um dos discos;
- Uso múltiplo de várias unidades de discos;
- Facilidade em recuperação de conteúdo perdido.

Existem duas formas de criarmos um RAID.

- **Via Software:** Feito por aplicativos e módulos do sistema operacional, o RAID via software só entra em funcionamento depois que o Kernel é carregado na memória do computador. A principal vantagem é a facilidade de configuração e a flexibilidade, já que podemos trabalhar com vários discos diferentes. A principal desvantagem é a dependência da correta configuração do sistema operacional.
- **Via Hardware:** Feito por uma placa controladora que conecta um disco ao outro. A principal vantagem é o desempenho, já que um RAID via hardware é mais rápido e independe do sistema operacional. A principal desvantagem, é que a placa controladora se torna um SPOF (Single Point of Failure), ou seja, é necessário ter uma controladora de discos igual ou compatível com a que você possui para o caso de falhas neste hardware.

3.2.1. Níveis de RAID

Os principais níveis de RAID utilizados hoje no mercado são os níveis 0,1, 5, e suas derivações, como por exemplo, o RAID 10. Vamos entendê-los:

- **RAID 0:** Este é o único nível de RAID que não implementa redundância. Sua finalidade é aumentar o desempenho de leitura e gravação, uma vez que ao gravar, divide os dados em partes iguais e armazena cada fragmento em um disco diferente simultaneamente. Por isso, com dois discos, a velocidade de leitura praticamente dobra. Com três discos, triplica. E assim por diante. Sua desvantagem é que se qualquer um dos disco falhar, o sistema operacional para de funcionar, além de ocasionar perda dos dados. São necessários ao menos dois discos para implementar RAID 0, e eles podem ser de tamanhos diferentes.
- **RAID 1:** O nível mais largamente utilizado. Sua principal finalidade é garantir redundância dos dados. A redundância é garantida pela duplicação dos discos que são igual e simultaneamente gravados em cada par de discos, logo, se um deles falhar, o outro continuará operando , até que a substituição do disco defeituoso seja feita. O ganho de desempenho está na leitura, uma vez que os

dados são lidos em partes iguais e simultaneamente de todos os discos.A desvantagem desse nível é que só metade do volume total de armazenamento nos discos utilizados ficará disponível para o sistema operacional. É preciso no mínimo dois discos para implementar RAID 1, sempre em pares.

- **RAID 5:** Neste nível de RAID teremos um balanço das vantagens e desvantagens do níveis anteriores, ou seja, RAID 5 provém um ganho de desempenho e tolerância a falhas a custos menores que RAID 0 ou RAID 1 individualmente. O ganho de desempenho está mais uma vez na leitura. Quanto mais discos forem adicionados a composição, mais rápida será a leitura, uma vez que a gravação é distribuídas em blocos de tamanho igual por todos os discos. A mágica do RAID 5 está justamente na divisão e distribuição destes blocos. Numa composição de três discos os dados serão divididos em dois blocos, A1 e B1, sendo que os bits destes dois blocos serão comparado através de um XOR ("ou exclusivo"). O resultado será gravado no terceiro volume como P1. Além disso, os blocos de paridade são alternadamente gravados em cada disco, aumentando a tolerância. Exemplo:

```
A1 B1 P1
A2 P2 C2
P3 B3 C3
```

Qualquer um dos discos que falhar pode ser rapidamente reconstruído através de novas operações XOR entre os dados restantes. Tomemos por exemplo A1 = 01001100 e B1 = 10100101.

```
01001100 XOR
10100101
--------
11101001
```

Agora suponha que o bloco B1 foi perdido. Para recuperá-lo basta aplicar um

XOR entre A1 e P1.

```
01001100 XOR
00010110
--------
01011010
```

A operação XOR significa que se são iguais "0 e 0" ou "1 e 1", então o resultado é verdadeiro (0). Se houver mais de dois blocos para serem comparados, calcule a paridade dos dois primeiros, e com resultado, compare com o terceiro, e assim sucessivamente. A principal desvantagem do RAID 5 é o custo de processamento da paridade. Portanto, RAID 5 é menos eficiente na gravação que seus antecessores.

A quantidade mínima de discos no RAID 5 é 3, podendo suportar qualquer número maior que 3, mesmo para discos de tamanhos diferentes. O volume disponível para armazenamento é dado pela equação (Q-1)*Dmenor, onde Q é a quantidade de discos, e Dmenor o tamanho do menor disco.

Os níveis de RAID podem ser combinados para se potencializar alguma das vantagens deles. As combinações mais comuns são os chamado RAID 10, RAID 0+1 e RAID 50.

A prova LPI pode cobrar conhecimentos do aluno sobre os níveis de RAID citados.

3.3. Prática dirigida

1) **Abra o particionador e marque as partições sda11, sda12 e sda13 sendo do tipo "RAID autodetect", caso não existam, crie:**

```
# cfdisk /dev/sda
```

2) **Verifique se o pacote mdadm já está instalado, e instale-o se necessário:**

```
# dpkg -l mdadm
# aptitude install mdadm
```

3) **Depois de instalar o pacote mdadm, abra um segundo terminal visualize o estado do RAID no Kernel:**

```
# watch cat /proc/mdstat
```

3.4. Criando o RAID

1) **Vamos criar o RAID utilizando o nível 1 utilizando "dois discos" e um "spare":**

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2 --spare-devices=1
/dev/sda11 /dev/sda12 /dev/sda13
```

No segundo terminal, verifique a sincronização do RAID

2) **Depois de criarmos o RAID, é necessário editar o arquivo /etc/mdadm/mdadm.conf, que será utilizado durante a administração do RAID:**

```
# vim /etc/mdadm/mdadm.conf
```

3) Modifique o arquivo, adicionando o seguinte conteúdo:

```
DEVICE        /dev/sda11 /dev/sda12 /dev/sda13
ARRAY         /dev/md0 devices=/dev/sda11,/dev/sda12,/dev/sda13
```

4) Depois de criarmos o RAID, aplique o sistema de arquivos ext3 no dispositivo /dev/md0:

```
# mkfs.ext3 /dev/md0
```

5) Crie um ponto de montagem em /mnt/raid

```
# mkdir /mnt/raid
```

6) Montando o RAID:

```
# mount -t ext3 /dev/md0 /mnt/raid
```

7) Configure o /etc/fstab:

```
/dev/md0 /mnt/raidext3defaults0
```

3.5. Verificando o funcionamento do RAID:

1) Verificando os dispositivos individualmente:

```
# mdadm -E /dev/sda11
# mdadm -E /dev/sda12
# mdadm -E /dev/sda13
```

2) **Para testar a redundância, vamos criar um script chamado "raidTest.sh" que escreve em um arquivo de 3 em 3 segundos:**

```
vim /root/TestaRaid.sh
```

```
#!/bin/bash
    while true ; do
    date >> /mnt/raid/dados.txt
    sleep 3
    done
```

```
chmod +x TestaRaid.sh
/root/TestaRaid.sh
```

3.6. Administrando o RAID

Vamos aprender a adicionar, remover e simular uma falha em um dos discos do RAID.

1) **Provocando uma falha:**

```
# mdadm /dev/md0 --fail /dev/sda11
```

2) **Verificando os detalhes do RAID após a falha:**

```
# mdadm --detail /dev/md0
```

3) **Removendo o disco defeituoso:**

```
# mdadm /dev/md0 --remove /dev/sda11
# mdadm --detail /dev/md0
```

4) Adicionando um disco ao RAID:

```
# mdadm /dev/md0 --add /dev/sda11
# mdadm --detail /dev/md0
```

5) Para parar o RAID:

```
# mdadm -S /dev/md0
```

6) Reiniciando o RAID:

```
# mdadm -As /dev/md0
```

A prova de certificação contempla o uso comum do comando mdadm bem como arquivos de configuração.

3.7. Exercícios Teóricos

1) Quais são as vantagens e desvantagens de usar o RAID 5?

____

____

____

____

2) Qual a diferença em se fazer um RAID via software e um RAID via hardware?

____

____

3) Quais são as vantagens e desvantagens de usar o RAID 1?

____

____

____

3.8. Laboratório

1. Utilizando os conhecimentos obtidos neste capítulo, crie um volume RAID-10;
2. Refazer todas as partições necessárias com o cfdisk, para que possamos testar o LVM no próximo capítulo;
3. Não esquecer de comentar as linhas dentro no /etc/fstab com a configuração do RAID, porque nosso sistema de salvamento de imagens precisa das partições padrões criadas.

Capítulo 4

LVM

4.1. Objetivos

- Criar novas partições no sistema com LVM;
- Aprender as diferenças entre volumes físicos, grupos de volumes e volumes lógicos;
- Criação de volumes físicos, grupos de volumes e volumes lógicos;
- Aplicar Sistemas de Arquivos nas partições criadas no LVM;
- Aumentar o tamanho das partições que utilizam LVM.

4.2. Introdução teórica

A sigla **LVM** significa: **Logical Volume Manager**. Ele é um gerenciador de discos que trabalha com camadas lógicas, que podem ser redimensionadas, aumentando ou diminuindo sem prejudicar o funcionamento do sistema.

A necessidade de se usar LVM é para que possamos aproveitar ao máximo o tamanho do HD ou de vários HD's. O LVM é bastante utilizado em sistemas virtualizados, onde há grande necessidade de várias partições.

4.3. Como funciona o LVM

O LVM é usado para agrupar vários discos de forma que o gerenciamento dos mesmos seja viável em um servidor de produção que não pode ficar desligado.

Vamos imaginar que temos 3 HD's em nossa máquina, e gostaríamos de aproveitar ao máximo de seus tamanhos.

Para não ter que se preocupar com o tamanho das partições logo de imediato, iremos usar o LVM para que possamos gerênciar os tamanhos das partições sem precisar fazer as famosas ``gambiarras''.

Para trabalhar com LVM as partições precisam estar formatadas com o tipo LVM. O **LVM trabalha com Grupos de Volumes** para alocar todas as partições que estão definidas como Volume Físicos do LVM. Esses volumes físicos serão divididos em vários Volumes lógicos como se fossem uma divisão de um disco (partição) para alocar o determinado ponto de montagem, e isso vai trazer a flexibilidade para redimensionar a determinada partição. Com mais detalhes vejamos as seguintes camadas que o LVM trabalha:

- **PV Physical Volume(Volume Físico)**: Os PV's são as partições que serão usadas para compor um disco no formato LVM, eles servem para dar o UUID, que é número de identificação de cada disco.

- **VG Volume Group(Grupo de Volume)**: Os VG's são um agrupamento de PV's, podemos ter vários VG's. Um exemplo é que podemos pegar a partições hda3 e hdb4 que estão declaradas como PV's e dizer que eles são o VG01, nisso podemos dividir esse grupo em vários pedaços com tamanhos específicos para ser associados a um diretório. Podemos utilizar todo o tamanho ou podemos deixar um espaço sobrando para quando precisar, aumentar os pedaços.

- **LV Logical Volume(Volume Lógico)**: Os LVs são os pedaços que falamos agora pouco, podemos classifica-los como sendo as partições de um disco, e VG sendo o disco. Eles são as partes que podem ser acessadas pelos usuários e que serão associados a um ponto de montagem específico. Um exemplo é falar que o LV01 será montado no diretório /home.

4.4. Definindo as Partições

Antes de criarmos as partições LVM, temos que ter em mente nossa tabela de particionamento:

```
# cat /etc/fstab
```

Agora iremos ver nossas partições:

```
# cat /proc/partitions
```

Por causa do capítulo de RAID iremos ter que recriar as partições novamente, e deixar estas Partições Livres:

Tipo	Device	Ponto de Montagem	Tamanho em MB	Filesystem
Log	/dev/sda10	Não montado	512 MB	Sem fs
Log	/dev/sda11	Não montado	512 MB	Sem fs
Log	/dev/sda12	Não montado	512 MB	Sem fs
Log	/dev/sda13	Não montado	512 MB	Sem fs
Log	/dev/sda14	Não montado	512 MB	Sem fs

Nas partições para realização deste trabalho, defina o tipo das partições como 8E, e utilize o cfdisk para realização desta tarefa. Não esqueça de reiniciar o computador.

```
# cfdisk /dev/hda
```

No final do capítulo, retire as linhas do /etc/fstab que iremos incluir e remova as configurações criadas com o LVM e RAID, para podermos salvar as imagens.

4.5. Prática Dirigida

1) **Verifique os pacotes necessários ao LVM, no Debian:**

```
# dpkg -l lvm2
```

Caso não tenha:

```
# aptitude install lvm2
```

2) **Red Hat: No Red Hat:**

```
# rpm -qa |grep lvm
```

3) **Caso não tenha:**

```
# yum install lvm2
```

4) **Gere o arquivo de configuração do LVM, utilize o comando vgscan.**

```
# vgscan
```

5) **Defina as novas partições como PV (Phisical Volume ou Volume Físico).**

```
# pvcreate /dev/sda11
# pvcreate /dev/sda12
```

6) **Verifique que as novas partições já estão utilizando LVM com sucesso:**

```
# pvscan
```

O pvscan, vai mostrar o dispositivo, o tamanho de cada partição, e a soma das duas partições.

7) Defina um Grupo de Volumes com os volumes físicos criados, nosso caso será o vg01:

```
# vgcreate vg01 /dev/sda11 /dev/sda12
```

8) Consultando o Grupo de Volumes (VG).

```
# vgdisplay
# vgdisplay -v vg01
```

9) Criando um Volume Lógico (LV) denominado teste:

```
# lvcreate -L 512mb -n lv01 vg01
```

10) Listando informações do Volume Lógico (LV):

```
# lvdisplay -v /dev/vg01/lv01
```

11) Consultando o Grupo de Volumes (VG). Repare que agora temos também os detalhes do "LV teste":

```
# vgdisplay -v vg01
```

12) Listando o device do LV criado:

```
# ls -l /dev/vg01/lv01
```

13) Verificando os LVs criados:

```
# lvs
```

14) Criando o sistema de arquivo EXT3 no Volume Lógico:

```
# mkfs -t ext3 /dev/vg01/lv01
```

4.6. Usando o LVM

1) Criando um ponto de montagem para LVM (caso seja necessário):

```
# mkdir /lvm
```

2) Montando manualmente o sistema LVM criado:

```
# mount -t ext3 /dev/vg01/lv01 /lvm
```

3) Verificando se o dispositivo está montado:

```
# df -h
# mount
```

Caso esteja utilizando RedHat (ou uma distribuição que siga o mesmo padrão),verifique a necessidade de criação de um label para o device. Se for necessário, faça-o. Red Hat: Com Label:

```
# e2label /dev/vg01/lv01 /lvm
```

Caso a máquina seja inicializada neste momento, o LVM não seria montado para utilização. Para que o volume LVM seja montado automaticamente, é necessário que seja editado o arquivo /etc/fstab e se adicione estas linhas no /etc/fstab:

4) Com label (para Red Hat):

```
LABEL=/lvm        /lvm       ext3         defaults     0      2
```

5) Sem label (para Debian):

```
/dev/vg01/lv01       /lvm       ext3         defaults      0      2
```

4.7. Administrando o LVM

1) Desmonte o LVM:

```
# umount /lvm
```

2) Redimencione o Volume Lógico (LV):

```
# lvextend -L +256mb /dev/vg01/lv01
```

3) Verifique o volume:

```
 # e2fsck -f /dev/vg01/lv01
```

Essa verificação é muito importante, faz a verificação no disco. Caso houver algum erro, é necessário fazer um "restauração".

4) Reestruturando o sistema de arquivos do Volume Lógico (LV):

```
# resize2fs /dev/vg01/lv01
```

5) Montando o LV:

```
# mount -t ext3 /dev/vg01/lv01 /lvm
```

6) Verificando a tabela de partições montadas:

```
# df -h
```

7) Copie alguns dados dentro do diretório /lvm:

```
# cp -r /etc /lvm
# cp -r /sbin /lvm
```

8) Verifique que os arquivos foram copiados corretamente:

```
# ls -lh /lvm
```

9) Verifique a tabela de partições:

```
# df -h
```

10) Desmonte o volume LVM:

```
# umount /lvm
```

4.8. Troubleshooting

Troubleshooting é a forma com a qual iremos resolver o problema.

1) Verifique o sistemas de arquivos:

```
# e2fsck -f /dev/vg01/lv01
```

2) Reestruturando o sistema de arquivos do Volume Lógico (LV):

```
# resize2fs -p /dev/vg01/lv01 256000
```

3) Reduzindo o LV. Em toda redução de espaço, há risco de perda de dados. Se quiser, para executar este procedimento, execute um backup da área de disco:

```
# lvreduce -L -256mb /dev/vg01/lv01
```

4) Verificando a tabela de partições montadas:

```
# df -h
```

5) Você ira perceber que o tamanho da LV realmente diminuiu.

```
# lvs
```

6) Montando o LV:

```
# mount -t ext3 /dev/vg01/lv01 /lvm
```

7) Copie alguns dados dentro do diretório /lvm:

```
# cp -r /bin /lvm
```

Verifique que os arquivos foram copiados corretamente:

```
# ls -lh /lvm
```

8) Desmonte o volume LVM:

```
# umount /lvm
```

9) Removendo o LV:

```
# lvremove /dev/vg01/lv01
```

Caso a partição esteja montada, não vai ser possível remover.

10) Após a remoção, verifique que não à mais o LV01:

```
# lvs
```

11) Removendo o grupo:

```
# vgremove vg01
```

12) O comando vgdisplay não ira mostrar mais nada:

```
# vgdisplay
```

13) Veja todas as suas Lvs:

```
# lvs
```

14) Verifique os detalhes do LVM:

```
# pvscan
```

4.9. Exercícios Teóricos

1) Como eu faço para estender uma partição criada com LVM?

____

____

2) Como eu faço para remover uma partição criada com LVM?

____

____

3) Eu posso aplicar um novo sistema de arquivos à partição da LVM, mesmo que ela contenha dados já criados nela?

____

____

4) Quando e por que, você utilizaria LVM em seu sistema?

____

____

5) Qual é o máximo de partições, que podemos ter em um DISCO, utilizando uma controladora IDE?

____

____

6) Qual é o arquivo que mostra todas as partições criadas na máquina?

____

____

7) Como eu consigo exibir a tabela de partições com os comandos: fdisk e cfdisk, sem entrar no programa?

____

____

8) Qual comando que eu consigo sincronizar, novas partições criadas em meu disco?

____

____

4.10. Laboratório

1. Pegue duas partições que não estão sendo utilizadas e crie um RAID1;
2. Adicione a partição como PV;
3. Verifique se estão todas funcionado;
4. Crie duas novas partições (LV) , a primeira 256mb e outra com 1GB;
5. Aplique o sistemas de arquivos swap na primeira e xfs na segunda;
6. Crie um ponto de montagem para a segunda LV com o nome de RESTORE;
7. Coloque as entradas no arquivo /etc/fstab;
8. Reinicie a máquina e verifique se está tudo bem;
9. Após todos os testes, remova tudo que foi feito com o RAID e LVM, inclusive suas entradas no /etc/fstab.

Capítulo 5

DHCP

5.1. Objetivos

- Entender o protocolo DHCP;
- Entender os métodos de atribuição de endereços;
- Configurar um servidor DHCP;
- Atribuir um endereço de acordo com MAC ADDRESS.

5.2. Introdução teórica

O DHCP **Dynamic Host Configuration Protocol** é um protocolo que que funciona nas camadas 2 e 3 do modelo OSI e é amplamente utilizado para oferecer endereçamento IP á um host que ainda não está configurado, o que oferece um flexibilidade ao Administrador de Sistemas. O protocolo DHCP opera da seguinte forma:

- **DHCPDISCOVER** - Um cliente envia um quadro broadcast (destinado a todas as máquinas) com um pedido DHCP;
- **DHCPOFFER** - O servidor DHCP captura o quadro e oferece um endereço IP ao cliente;
- **DHCPREQUEST** - O cliente envia um DHCP REQUEST endereçado para o servidor DHCP aceitando o IP;
- **DHCPACK** - Esse é o pacote que confirma a atribuição de uma configuração de

rede a um cliente, ou seja, aquele cliente agora possui configurações distribuídas pelo servidor DHCP;

- **DHCPNAK** - Caso o cliente não aceite aquele endereço IP, ele enviará um DHCPNAK para o servidor, e realizará o DHCPDISCOVER novamente.

O DHCP oferece três tipos de alocação de endereços IP:

- **Atribuição manual** - Quando desejamos que certo cliente tenha determinado endereço IP, temos que "amarrar" o endereço MAC da máquina do cliente no endereço IP desejado. O servidor de DHCP descobre o MAC ADDRESS do cliente através do DHCPDISCOVER, assim identificando quais são as máquinas que irão receber configurações personalizadas;
- **Atribuição automática** - Onde o cliente obtém um endereço de um espaço de endereços possíveis chamado de range, especificado pelo administrador. Geralmente não existe vínculo entre os vários MAC's habilitados a esse espaço de endereços. Assim que o cliente se loga pela primeira vez na rede, ele recebe um endereçamento definitivo;
- **Atribuição dinâmica** - O único método que dispõe a reutilização dinâmica dos endereços. O administrador disponibiliza um espaço de endereços possíveis, e cada cliente terá o software TCP/IP da sua interface de rede configurados para requisitar um endereço por DHCP assim que a máquina inicialize. A alocação utiliza um mecanismo de aluguel do endereço, caracterizado por um tempo de vida. Após a máquina se desligar, o tempo de vida naturalmente irá expirar, e da próxima vez que o cliente se ligue, o endereço provavelmente será outro.

Algumas implementações do software servidor de DHCP permitem ainda a atualização dinâmica dos servidores de DNS para que cada cliente disponha também de um DNS. Este mecanismo utiliza o protocolo de atualização do DNS especificado no RFC 2136.

5.3. Prática dirigida

5.3.1. Configurando um servidor DHCP

1) Instalar o pacote do servidor dhcp3:

```
# aptitude install dhcp3-server
```

2) O arquivo de configuração é o dhcpd.conf no exemplo abaixo, e fica dentro de /etc/dhcp3:

```
# vim /etc/dhcp3/dhcpd.conf
```

Onde X, é a rede que o professor informar:

```
1 ddns-update-style none;
2           subnet 192.168.X.0 netmask 255.255.255.0 {
1           range 192.168.X.10 192.168.X.200;
2           option subnet-mask 255.255.255.0;
3           option domain-name "microx.com.br";
4           option domain-name-servers 200.176.2.10,4.2.2.2;
5           option routers 192.168.X.254;
6           default-lease-time 600;
7           max-lease-time 7200;
8 }
```

A prova de certificação pode cobrar a utilização de cada opção, como "option routers" ou "option domain-name-servers".

3) Uma vez criado o arquivo de configuração, basta inicializar o servidor DHCP:

```
# invoke-rc.d dhcp3-server stop
# invoke-rc.d dhcp3-server start
```

5.3.2. Principais diretrizes do arquivo de configuração

- **default-lease-time 600** - Servidores DHCP cedem endereços sob pedido por um tempo pré-determinado. O padrão nesse exemplo é ceder o endereço IP por 600 segundos, ou 10 minutos;
- **max-lease-time 7200** - Caso o cliente solicite um tempo maior, o tempo máximo permitido será de 7.200 segundos (2 horas);
- **option subnet-mask 255.255.255.0** - Essa opção define a máscara de subrede a ser fornecida aos clientes;
- **option broadcast-address 192.168.200.255** - Essa opção define o endereço de envio para requisições de broadcast;
- **option routers 192.168.200.254** - O cliente, além do número IP, recebe também a informação do número do host que é o gateway de sua rede;
- **option domain-name-servers 200.176.2.10,4.2.2.2** - Essa opção lista os servidores de nomes (DNS) a serem utilizados para resolução de nomes;
- **option domain-name microx.com.br** - O nome de domínio do cliente.

5.4. Configurando os clientes DHCP

1) Do lado cliente, temos duas opções para fins de teste:

```
# dhclient
```

Caso não tenha o comando dhclient:

 # aptitude install dhcp3-client

2) Ou editando o arquivo /etc/network/interfaces, trocando static por dhcp, ficam assim:

```
# vim /etc/network/interfaces
```

```
auto eth0
iface eth0 inet dhcp
```

Para visualizar a placa de rede que está utilizando:

 # mii-tool

O arquivo dos leases do dhcp se localiza em /var/lib/dhcp3/dhcpd.leases.

3) Este é o arquivo onde ficam registrados os empréstimos de IP's. Observe-o:

```
# more /var/lib/dhcp3/dhcpd.leases
```

5.5. Fixar IP via DHCP

É possível fixar o IP via DHCP para máquinas respectivas. Para isso, precisamos associar o MAC ADDRESS da placa com um IP.

1) Considere que:

MÁQUINA	*MAC ADRESS*	*IP FIXADO*
micro 1	00:80:C7:D2:F8:D5	192.168.200.210
micro 2	88:3D:BE:00:C7:00	192.168.200.214

2) Para esse cenário ser possível, seria necessária a respectiva entrada no arquivo de configuração /etc/dhcp3/dhcpd.conf:

```
3  host micro1 {
1         hardware ethernet 00:80:C7:D2:F8:D5;
2         fixed-address 192.168.200.210;
3  }
4
5  host micro2 {
6         hardware ethernet 88:3D:BE:00:C7:00;
7         fixed-address 192.168.200.214;
8  }
```

3) O Arquivo /etc/dhcp3/dhcpd.conf ficaria assim:

```
4 ddns-update-style none;
5
1      subnet 192.168.X.0 netmask 255.255.255.0 {
2      range dynamic-bootp 192.168.200.1 192.168.200.200;
3
4      option routers192.168.X.254;
5      option subnet-mask 255.255.255.0;
6      option domain-name microx.com.br;
7      option domain-name-servers200.204.0.10, 200.204.0.138;
8      default-lease-time 21600;
9      max-lease-time 43200;
10
11  host micro1 {
12     hardware ethernet 00:80:C7:D2:F8:D5;
13     fixed-address 192.168.X.210;
14    }
15
16  host micro2 {
17     hardware ethernet 88:3D:BE:00:C7:00;
18     fixed-address 192.168.X.214;
19    }
20  }
```

5.6. Exercícios teóricos

1) Um servidor DHCP precisa ser necessariamente o gateway da rede? Explique.

____

____

____

____

2) Podemos ter mais de um servidor de DHCP em uma rede? Explique.

____

____

____

____

3) Quais são as três formas de alocação endereçamento IP?

____

____

____

____

5.7. Laboratório

1. Retire o daemon de inicialização do DHCP do runlevel;
2. Reinicie a máquina e veja se o serviço do DHCP está inativo.

Capítulo 6

FTP

6.1. Objetivos

- Instalar e configurar um servidor FTP;
- Entender as diferenças entre FTP passivo e ativo;
- Utilizar comandos de FTP para download e upload;
- Permitir a utilização do FTP com usuário anônimo.

6.2. Introdução teórica

O FTP File Transfer Protocol é um protocolo simples para transferência de arquivos. O cliente FTP faz uma solicitação ao servidor FTP, a seção é estabelecida e então é solicitado o usuário e senha válidos no caso de um FTP autenticado, ou, caso este servidor permita navegação anônima, basta entrar como o usuário "anonymous" e um endereço de e-mail qualquer como senha. O FTP pode atuar como servidor ativo ou passivo.

No modo ativo, os comandos são enviados por uma porta alta pelo cliente, e são recebidas pela porta 21 no servidor, enquanto que os dados são transmitidos pelo servidor ao cliente através da porta 20. O problema desta implementação é que os dados podem ser barrados por um Firewall de acordo com as regras estabelecidas pelo Administrador de Sistemas.

Já no modo passivo, os comandos também são enviados para o servidor através de uma porta alta pelo cliente, e são recebidas na porta 21 do servidor. Neste momento, o cliente avisa ao servidor que ele deve utilizar o modo passivo através do comando "PASV", e então os dados serão enviados utilizando portas altas tanto pelo cliente quando pelo servidor. Neste caso, não temos mais problemas com o firewall no lado do cliente, porém, temos que habilitar a utilização de portas altas no servidor, o que pode gerar muitos problemas. Felizmente, na configuração do servidor FTP podemos especificar o range de portas que o servidor deve utilizar, minimizando assim o problema no lado do servidor.

Sobre segurança, um dos principais problemas do FTP é que a maioria dos servidores não implementa criptografia, então, caso você deseja um ambiente seguro com FTP, é necessário a implementação de criptografia, como OpenSSL ou TLS.

6.3. Prática dirigida

6.3.1. Servidor FTP

1) Verifique se você possui o servidor proftpd instalado em seu sistema:

```
# dpkg -l proftpd
```

2) Vamos instalá-lo com o aptitude:

```
# aptitude install proftpd
```

Red Hat:
A plataforma Red Hat utiliza o vsftpd por padrão.

3) Para configurar o seu servidor FTP, edite o arquivo de configuração e altere as diretivas listadas a seguir:

```
# vim /etc/proftpd/proftpd.conf
```

4) Para trabalhar em modo standalone:

```
ServerType    standalone
```

5) Defina o valor padrão de UMASK para gravação:

```
Umask         022  022
```

6) Defina o número máximo de logins simultâneos:

```
MaxInstances  20
```

7) Habilite até 5 conexões de usuários anonymous:

```
MaxClients    5
```

Cada usuário do FTP pode ter uma mensagem de login diferente. Crie o arquivo welcome.msg na home do usuário.

8) Você pode usar o arquivo de boas vindas para logins anônimos /home/ftp/welcome.msg como base:

```
# cp /home/ftp/welcome.msg /home/aluno
```

Edite o arquivo copiado:

```
# vim /home/aluno/welcome.msg
```

Inicie o serviço do Proftpd:

```
# invoke-rc.d proftpd stop
# invoke-rc.d proftpd start
```

9) Verifique em qual porta o servidor FTP está escutando:

```
# netstat -nltup
```

6.4. Conectando-se a um servidor FTP como cliente

Um servidor FTP pode ser usando de duas formas:

- **Tradicional** - Neste formato, o servidor aceita conexões de um usuário e senha válidos para liberar um shell para ele.
- **Anonymous** - O Servidor FTP com anonymous é muito utilizado na Internet pelo motivo de não ser necessário ter um usuário no servidor. Desta forma, o usuário pode abrir um browser e chamar o endereço ftp://servidor para ter acesso ao diretório disponibilizado pelo serviço. Geralmente, esse diretório é o home do usuário FTP que no Debian é /home/ftp.

1) Conecte-se ao servidor FTP do colega ao lado, fornecendo o nome de usuário aluno e a senha padrão:

```
# ftp 192.168.200.X
```

2) Verifique se a conexão foi bem sucedida e encerre a sessão.

```
ftp> quit
```

3) Agora conecte-se como um usuário anonymous, fornecendo um e-mail qualquer como senha.

```
# ftp 192.168.200.x

Connected to localhost.
220 ProFTPD 1.3.0 Server (Debian) [::ffff:127.0.0.1]Name
(localhost:aluno): anonymous
331 Password required for anonymous.
Password:
```

4) A conexão foi recusada porque o padrão do Proftpd é não aceitar conexões anônimas. Vamos habilitar a navegação anônima. :)

```
# vim /etc/proftpd.conf
```

Tire um comentário por linha, da linha 132 até o final do arquivo:

```
# <Anonymous ~ftp>
136 #User ftp
137 #Group nogroup
138 ## We want clients to be able to login with "anonymous" as well as
"ftp"
139 #UserAlias anonymous ftp
140 ## Cosmetic changes, all files belongs to ftp user
141 #DirFakeUser on ftp
142 #DirFakeGroup on ftp
143 #
144 #RequireValidShell off
145 #
146 ## Limit the maximum number of anonymous logins
147 #MaxClients10
...
Até o final do arquivo retirar um # comentário por linha.
```

5) Reinicie o serviço para que as alterações tenham efeito:

```
# invoke-rc.d proftpd restart
```

6) Tente novamente a conexão como usuário anônimo. Agora deverá funcionar. Veja o quadro abaixo:

```
# ftp 192.168.200.X
```

```
Name (localhost:aluno): anonymous
Password:
```

Continue logado como usuário anônimo. Iremos fazer alguns testes a seguir.

6.5. Comandos FTP

Os servidores de FTP muito raramente mudam, mas novos programas clientes FTP aparecem com bastante regularidade. Estes clientes variam no número de comandos que implementam. A maioria dos clientes FTP comerciais implementam apenas um pequeno subgrupo de comandos FTP. Mesmo que o FTP seja um protocolo orientado à linha de comandos, a nova geração dos clientes FTP esconde esta orientação num ambiente gráfico muitas vezes bastante desenvolvido.

As interfaces clientes do FTP do BSD UNIX e do GNU/Linux possuem muitos comandos, alguns deles arcaicos e sem utilidade hoje em dia, como por exemplo o tenex e o carriage control. Já outros são bastante utilizados: cd, dir, ls, get, mget, put e mput.

Os comandos listados abaixo podem ser cobrados na prova de certificação.

Abaixo estão listados alguns dos mais utilizados comandos FTP:

- **help** - Lista os comandos disponíveis. Um sinônimo é ?
- **help CMD** - Mostra uma ajuda para o comando CMD
- **ls** - Lista os aquivos no servidor. Um sinônimo é dir
- **cd** - Troca de diretório no servidor
- **lcd** - Troca de diretório da máquina local
- **!ls** - Lista os arquivos da máquina local
- **!CMD** - Executa na máquina local o comando CMD
- **get** - Faz download de um arquivo do servidor para a máquina local.
- **mget** - Faz download de mais de um arquivo.
- **put** - Faz upload de um arquivo da máquina local para o servidor.
- **mput** - Faz upload de mais de um arquivo.

Então como usuário anônimo, vamos fazer alguns testes:

1) Liste o conteúdo do servidor:

```
ftp> ls
```

2) Liste o conteúdo do seu diretório local:

```
ftp> !ls
```

3) Faça o download de algum arquivo:

```
ftp> get ARQUIVO
```

4) Verifique se o arquivo foi copiado:

```
ftp> !ls
```

5) Tente agora fazer o upload de um arquivo:

```
ftp> put ARQUIVO
```

Por padrão, usuários anônimos não devem ter permissão para fazer upload de arquivos.

6) Encerre a sessão e logue-se em seguida como um usuário válido no servidor FTP da máquina de algum colega:

```
ftp> quit
221 Goodbye.

# ftp 192.168.200.X
```

Tente agora fazer o upload do arquivo. Deverá funcionar desta vez.

7) Faça o download de vários arquivos:

```
ftp> mget *
```

Note que é exigida a confirmação para cada arquivo copiado, o que pode ser incômodo.

8) Desligue o modo interativo, e tente novamente:

```
ftp> prompt
Interactive mode off.
ftp> mget *
```

9) Faça o upload de vários arquivos:

```
ftp> mput *
```

10) Verifique o log de atividade do FTP:

```
# less /var/log/proftpd/xferlog
```

6.6. Exercícios teóricos

1) Para que serve a porta ftp-data?

___

2) Qual a diferença dos modos passivo e ativo?

___

___

3) Em termos de segurança, qual a maior falha do FTP?

___

6.7. Laboratório

1. Permita que o usuário anonymous poste dados em um diretório chamado incoming;

Capítulo 7

NFS

7.1. Objetivos

- Instalar o daemon NFS;
- Entender a configuração do NFS;
- Exportar um diretório usando NFS.

7.2. Introdução teórica

O NFS Network File System, é um sistema de arquivos especial capaz de exportar um diretório via rede.

Ao importar um diretório, a impressão que o cliente tem é de que o diretório está localizado no próprio computador, o que torna o acesso e utilização do diretório transparente para o usuário final, o que torna o NFS uma solução interessante para centralização de diretórios pessoais e recursos compartilhados em rede, já que as operações de backup e manutenção serão centralizadas.

Para que os clientes possam acessar o servidor NFS é necessário que os seguintes serviços estejam sendo executados no servidor:

- **nfsd** - Serviço NFS que atende as requisições dos clientes NFS;
- **mountd** - Serviço que executa as solicitações de montagem dos clientes NFS;
- **portmap** - Serviço que permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

7.3. Prática Dirigida

7.3.1. Instalação e configuração do NFS

SERVIDOR:

1) No Debian, basta instalar o pacote nfs-kernel-server que os outros serão instalados:

```
# aptitude install nfs-kernel-server
```

2) Agora podemos determinar quais diretórios pretendemos compartilhar, editando o arquivo /etc/exports:

```
# vim /etc/exports
```

3) Sintaxe do arquivo:

```
/diretório_compartilhado
    192.168.200.0/24(rw,no_root_squash,subtree_check)
/outro_diretório        192.168.100.1(ro,root_squash)
```

A prova de certificação pode conter perguntas sobre a configuração de um servidor NFS

Note as opções root_squash e no_root_squash:
A opção root_squash não permite que o root local tenha poderes administrativos dentro do diretório compartilhado, ou seja, é como se o root "perdesse os poderes" sobre qualquer operação naquele diretório.

Já a opção no_root_squash permite que o root local administre aquele diretório, podendo causar sérios problemas, já que o root local terá poderes administrativos em um diretório compartilhado, podendo apagar arquivos de outros usuários, ler documentos e mudar as permissões.

4) Crie o diretório /srv/nfs para ser exportado e em seguida, adicione alguns arquivos nele:

```
# mkdir -p /srv/nfs
# cp /etc/apt/* /srv/nfs
```

5) Edite o arquivo /etc/exports para que o diretório criado seja exportado:

```
/srv/nfs 192.168.200.0/24(rw,root_squash)
```

6) Reinicie o serviço NFS:

```
# invoke-rc.d portmap restart
# invoke-rc.d nfs-kernel-server restart
```

7) Verificando em quais portas locais o nfs está trabalhando:

```
# rpcinfo -p localhost
programa versão protocoloporta
100000 2tcp 111 portmapper
100024 1udp 722 status
100003 2udp2049 nfs
```

CLIENTE:

1) **Na máquina cliente, basta instalarmos o pacote nfs-client:**

```
# aptitude install nfs-common
```

2) **Verifique todos os diretórios compartilhados pelo colega através do comando:**

```
# showmount -e 192.168.200.X
```

3) **Crie um ponto de montagem:**

```
# mkdir /mnt/nfs
```

4) **Monte o diretório compartilhado:**

```
# mount -t nfs 192.168.200.X:/srv/nfs /mnt/nfs
```

5) **Podemos verificar todas as conexões com o servidor:**

```
# showmount -a 192.168.200.X
```

6) **Visualize também com o comando mount os pontos de montagem :**

```
# mount
```

7) **Explore o diretório remoto:**

```
# ls /mnt/nfs
```

8) **Como administrador, tente remover algum arquivo do diretório remoto:**

```
# cd /mnt/nfs
# rm ARQUIVO
```

SERVIDOR:

Como informado, por padrão, o NFS assume a opção root_squash que não reflete os direitos de administrador aos clientes. Vamos alterar isso e fazer alguns testes.

1) **Edite o arquivo /etc/exports e acrescente:**

```
/srv/nfs 192.168.200.0/24(rw,no_root_squash)
```

2) **Não é necessário reiniciar o NFS. O comando exportfs permite gerenciar os diretórios compartilhados de forma dinâmica:**

```
# exportfs -r
```

A opção -r faz com que o arquivo /etc/exports seja relido. Consulte a manpage do exportfs para saber sobre outras opções que permitem exportar novos diretórios sem a necessidade de alterar o arquivo /etc/exports ou reiniciar o NFS.

CLIENTE:

Agora, na máquina cliente tente novamente remover algum arquivo.

1) **Faça este teste como administrador:**

```
# cd /mnt/nfs
# rm ARQUIVO
```

Observe que não há a necessidade de remontar o sistema de arquivos compartilhado, toda alteração foi feita de forma dinâmica.

7.4. Exercícios teóricos

1) Para que serve a opção no_root_squash no arquivo /etc/exports?

2) Qual tipo de filesystem deve ser utilizado com o NFS?

3) Você centralizou os diretórios pessoais em um servidor. Como você deve configurar o cliente para que o compartilhamento seja montado durante o boot?

7.5. Laboratório

1. Crie um mesmo diretório com permissão de escrita para um IP e somente leitura para um outro IP. Exporte esse diretório através do NFS;
2. Crie um compartilhamento chamado publico e exporte-o para a rede 192.168.200.0/24.

Capítulo 8

Servidor SAMBA

8.1. Objetivos

- Entender as funcionalidades do SAMBA;
- Entender seu arquivo de configuração;
- Criar compartilhamento de diretórios;
- Criar um servidor de autenticação.

8.2. Introdução teórica

A suíte SAMBA começou a ser desenvolvida por Andrew Tridgell em 1992 como ferramenta para compartilhamento de diretórios e arquivos entre máquinas *nix e maquinas com sistema operacional Windows e OS/2, da IBM.

Utilizando engenharia reversa no protocolo SMB, Server Message Block, Andrew foi capaz de implementar algumas funcionalidades SMB em máquinas *nix. Mais tarde, o IETF batizou este conjunto de funcionalidades como CIFS, Common Internet File System. Além de promover esta interoperabilidade, o SAMBA também é capaz de realizar algumas tarefas citadas abaixo:

- Compartilhar um ou mais tipos de sistemas de arquivos;
- Compartilhar impressoras em uma rede NT, ou atuar como cliente;
- Autenticação de clientes em um domínio Windows.

A suíte SAMBA precisa de três componentes para realizar sua função. São eles:

- **nmbd** - Responsável pela resolução de nomes
- **smbd** - Responsável por compartilhar recursos
- **winbind** - Auxilia na autenticação em um domínio AD

8.3. Prática Dirigida

1) Vamos instalar a suíte SAMBA, algumas ferramentas auxiliares e o pacote de documentação:

```
# aptitude install samba samba-doc smbclient smbfs
```

2) Veja o arquivo de configuração, e tenha como hábito sempre guardar uma cópia do arquivo original:

```
# vim /etc/samba/smb.conf
# mv /etc/samba/smb.conf /etc/samba/smb.conf.original
```

Red Hat:

Na plataforma Red Hat, o arquivo de configuração do samba fica em /etc/smb.conf

Iremos agora gerar um novo arquivo /etc/samba/smb.conf com as opções que comentamos acima:

```
[global]
    workgroup = microx
    server string = Servidor de Arquivos
    security = SHARE
    wins support = Yes

[Publico]
    comment = Diretorio Publico
    path = /srv/samba/publico
    force user = smbuser
    force group = users
    read only = No
    guest ok = Yes
```

Correspondem:

```
[global]
    workgroup = Grupo de Trabalho;
    server string = Comentário para o servidor;
    security = Compartilhamento sem a necessidade de controle de
usuários e  senhas;
    wins support = O samba se torna um Servidor Wins, resolve nome
para máquinas Windows.

[Publico]
    comment = Adicionar comentário ao compartilhamento;
    path = Diretório que será utilizado para compartilhamento;
    force user = Define um usuário padrão que será usado por todos que
acessarem este compartilhamento;
    force group = Define um grupo padrão que será usado por todos que
acessarem este compartilhamento;
    read only = Permissões de leitura e gravação;
    guest ok = Define que usuários “convidados” terão acesso ao
compartilhamento.
```

3) O SAMBA oferece um comando para verificar a sintaxe do arquivo de configuração. Vamos utilizá-lo:

```
# testparm
```

4) Nós definimos que todo arquivo criado no compartilhamento seria do usuário smbuser e do grupo users. Vamos criar este usuário e alterar seu grupo:

```
# useradd smbuser -g users
```

5) Também é necessário criar o diretório que será compartilhado:

```
# mkdir -p /srv/samba/publico
# chown smbuser:users /srv/samba/publico
```

6) Reinicie o SAMBA para que as alterações tenham efeito:

```
# invoke-rc.d samba stop
# invoke-rc.d samba start
```

7) Verifique se os daemons (smbd e nmbd) estão atendendo requisições:

```
# netstat -putan
```

8) Verifique o compartilhamento:

```
# smbclient -L localhost
```

9) Verifique o compartilhamento da máquina ao lado:

```
# smbclient -L 192.168.200.X
```

10) Se tudo ocorreu bem, podemos montar o compartilhamento, escolha uma das formas abaixo:

```
# smbmount //192.168.200.X/Publico /mnt
# mount -t cifs //192.168.200.X/Publico /mnt
```

11) Confirme se o compartilhamento foi montado, e então crie um arquivo no compartilhamento:

```
# mount
# touch /mnt/$HOSTNAME.txt
```

12) Verifique as permissões do arquivo criado pelo colega:

```
# ls -la /srv/samba/publico
```

8.4. Samba como controlador de Domínio Primário

Agora que já estamos mais familiarizados com o SAMBA, podemos configurá-lo para atuar como um Controlador de Domínio Primário.

.

Configuração para PDC

A maior vantagem de ter um controlador de domínios Samba é ter a estabilidade e flexibilidade do Linux controlando os logins em uma rede Microsoft.

Para que isso seja possível, nós iremos editar o nosso arquivo /etc/samba/smb.conf de forma o Samba fique apto a gerenciar os logins e computadores da rede.

Edite o seu arquivo /etc/samba/smb.conf para que ele fique como o exemplo abaixo:

```
1 [Global]
2         netbios name = SERVER
3         workgroup = EMPRESA
4         server string = Primary Domain Controller
5         log file = /var/log/samba/%m.log
6         max log size = 100
7         security = user
8         unix password sync = Yes
9         passwd program = /usr/bin/passwd %u
10        smb passwd file = /etc/samba/smbpasswd
11        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
12        domain logons = Yes
13        os level = 100
14        preferred master = Yes
15        domain master = Yes
16        local master = yes
17        logon drive = H:
18        logon home = \\%L\homes\%U
19        logon path = \\%L\Profiles\%U
```

13) Agora, vamos criar o compartilhamento homes que irão armazenar os dados pessoais dos usuários:

```
1 [homes]
2    comment = Diretorio Pessoal
3    path = /srv/samba/homes/%U
4    valid users = %U
5    read only = No
6    browseable = No
```

1. Compartilhamento homes;
2. Comentário;
3. Localização do compartilhamento no servidor;
4. Usuários que podem acessar o compartilhamento (%U = usuário da seção);
5. Somente leitura desativado;

6. Não permite visualização por outros usuários.

Esteja atento a configuração de compartilhamentos e suas opções comuns!

14) Por ultimo, iremos criar os Perfis Móveis, que armazenarão as configurações do usuário:

```
7 [Profiles]
8     comment = Perfis Moveis
9     path = /srv/samba/profiles
10    read only = No
11    guest ok = Yes
12    browseable = No
```

1. Compartilhamento Profiles;
2. Comentário;
3. Localização do compartilhamento, no servidor;
4. Somente leitura desativado;
5. Habilitando perfil móvel para usuário convidado;
6. Não permite visualização por outros usuários.

Com essas configurações, o SAMBA está pronto para atuar como controlador de domínios, porém, será necessário cadastrar os usuários e computadores Windows no samba, criar as contas de usuários e acertar as permissões dos usuários também. Vamos ver isso na próxima seção.

15) Vamos manter o Compartilhamento Público aberto para todos:

```
[Publico]
      path = /srv/samba/publico
      browseable = yes
      writeable = yes
      public = yes
```

8.5. Cadastrando usuários no PDC

Para que nossos usuários possam Windows possam efetuar login no domínio, é necessário que tanto o host quanto o usuário estejam cadastrados na conta normal do computador e também na conta SAMBA. Vamos configurar isto.

1) O usuário root é quem poderá adicionar uma máquina a um domínio samba:

```
# smbpasswd -a root
```

2) Antes de prosseguirmos, é necessário criar os diretórios:

```
# mkdir /srv/samba/profiles
# mkdir /srv/samba/homes
```

3) E alterar suas permissões:

```
# chmod 775 /srv/samba/profiles
# chmod 775 /srv/samba/homes
# chown root:users /srv/samba/profiles
```

4) Também precisamos criar o usuário que irá autenticar-se no SAMBA:

```
# useradd -c "Seu Nome Completo" -m -d /srv/samba/homes/microx -g
users -s /bin/false microx
```

5) Atribuir a senha para o novo usuário:

```
# smbpasswd -a microx
```

6) E cadastrar a máquina que entrará no domínio:

```
# useradd -c "HOSTNAME" -d /dev/null -s /bin/false hostname$
```

Fique atento com a LPI, ele pode cobrar os nomes dos modulo de autenticação do samba:

/etc/pam.d/common-auth

auth required /lib/security/pam_winbind.so
account required /lib/security/pam_winbind.so

8.6. Exercícios Teóricos

1) O fazem os serviços nmbd e smbd?

____

____

2) O que é um Roaming Profile e quais as suas vantagens?

____

____

8.7. Laboratório

1. Implemente um compartilhamento público;
2. Implemente o Samba com PDC.

Capítulo 9

Domain Name System

9.1. Objetivos

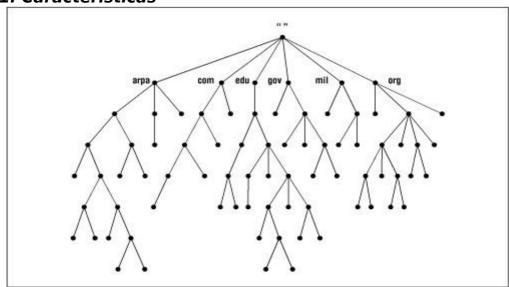
- Entender sobre resolução de nomes;
- Realizar consultas DNS;
- Configurar uma zona de domínio DNS;
- Configurar uma zona reversa de domínio DNS.

9.2. Introdução teórica

No final dos anos 70, a introdução do protocolo TCP/IP e conseqüentemente a rápida expansão da ARPAnet tornou obsoleto o sistema de atualização manual e centralizado do arquivo HOSTS.TXT, que continha uma tabela associando todas informações sobre os hosts da ARPAnet, incluindo seus endereços. Era simples mantê-lo para poucas centenas de máquinas, mas tornou-se impraticável para milhares e milhares que se conectavam a rede em um processo exponencial de crescimento. Diante desse problema, a direção da ARPAnet contratou pesquisadores para desenvolver uma solução que atendesse às seguintes especificações:

- Permitir administração local e ao mesmo tempo publicação global;
- Garantir univocidade do nome do hosts através de uma distribuição hierárquica de domínios.

Em 1984 foi liberado o Domain Name System, descritos nas RFC's 882 e 883. Atualmente estas RFC's foram suplantadas pelas de números 1034 e 1035, além de RFC's suplementares de segurança, administração, atualização dinâmica de servidores de nome, e muitas outras.

9.2.1. Características

- Banco de dados hierárquico e distribuído representado no formato de uma árvore invertida e 127 níveis;
- Namespace de até 63 caracteres;
- É capaz de associar outras informações a um host e não só seus endereços IP's;
- Arquitetura cliente/servidor;
- Os clientes são chamados resolvers, e costumam ser bibliotecas do sistema operacional (libresolv no linux) compartilhadas entre os mais diversos programas, como o ping ou o navegador web;
- Do outro lado estão os servidores de nome DNS (DNS nameserver);
- A raiz da árvore tem nome nulo ou "", por isso a representamos simplesmente como ponto (.);
- Os nós abaixo do domínio raiz são chamados domínios de nível mais elevado (top level domains);
- Sua quantidade e nomes são impostos pela ICANN (Internet Corporation for

Assigned Names and Numbers);

- Eles são divididos em gTLD e ccTLD, onde temos respetivamente o domínios genéricos com, edu, gov, mil, etc; e os códigos de países (country-code), sempre com duas letras;
- A ICANN delega, de acordo com tratados internacionais, a responsabilidade pela administração de um ccTLD;
- No caso do Brasil, essa responsabilidade pertence atualmente ao CGI.br, mais especificamente ao REGISTRO.br;
- Uma vez delegado um domínio, sua nova autoridade pode delegar subdomínios sem necessitar notificar a entidade responsável pelo domínio pai;
- Um subdomínio está para um subdiretório assim como um domínio está para um diretório, e um host está para um arquivo;

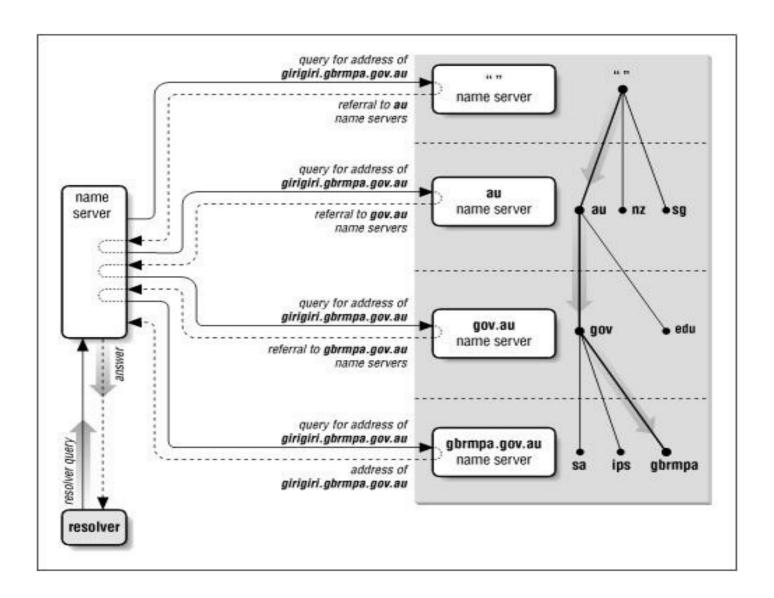
Finalmente, vale a pena mencionar que o arquivo HOSTS.TXT foi portado para ambiente Unix (e posteriormente Linux) como /etc/hosts. Este arquivo é normalmente o primeiro a ser consultado pelo resolvedor, que irá buscar por um servidor de nomes apenas em caso de o host não ser encontrado no arquivo /etc/hosts.

9.3. Resolução

Solução DNS é o processo pelo qual um programa consulta dados a respeito de um hostname. Na grande maioria das vezes, consulta-se o endereço IP deste host, para então efetuar algum tipo de conexão à algum serviço, como HTTP, SMTP. POP, dentre inúmeros outros.

O processo de resolução, a partir do primeiro nameserver consultado, pode ser:

- **recursiva**
- **iterativa**

9.3.1. Resolução Recursiva

Tomando um navegador web como exemplo, a resolução para acesso a um website tem as seguintes etapas:

1. Usuário solicita acesso a www.exemplo.com.br;

2. Navegador checa se já conhece o endereço IP do hostname solicitado (cache do browser);

3. Se não conhece, o navegador passa a solicitação para a biblioteca de resolução (resolver);

4. O resolver procura o hostname solicitado no arquivo /etc/hosts local;

5. Se não encontrar, ele checa o arquivo /etc/resolv.conf para saber a quais nameservers deve solicitar a informação;

6. O resolver repassa a solicitação ao primeiro nameserver da lista, e logo após para o próximo até o fim da lista, aguardando por uma resposta de qualquer um deles;

7. O servidor de nomes acionado consulta seu cache, se houver;

8. Se não encontrar em seu cache, o servidor em questão vai diretamente ao servidor raiz e transfere a consulta (www.exemplo.com.br?);

9. O servidor raiz não faz cache, e também não é autoridade sobre zonas de baixo nível, então ele apenas responde uma parte da questão: "Não sei quem é, mas sei quem pode responder melhor: br.";

10. O servidor de nomes reenvia a consulta para o servidor .br. (www.exemplo.com.br?);

11. .br retorna o mesmo tipo de resposta, porém como uma dica mais próxima: "Não sei quem é, mas sei quem pode responder melhor: com.br.";

12. Passo 10 e 11 são efetuados mais uma vez, e agora a resposta é "Não sei quem é, mas sei quem pode responder melhor: exemplo.com.br.";

13. Após repetir o passo 10, finalmente a resposta será da autoridade sobre o domínio exemplo.com.br. Vai ser respondido o IP, juntamente ao TTL do registro, ou será respondido "inexistente";

14. O servidor de nomes fará cache da resposta, ao mesmo tempo que a repassa para o resolvedor original;

15. O resolvedor repassa a resposta para o navegador;

16. 0 navegador inicia uma conexão HTTP com o IP descoberto.

Conceitos de DNS e a configuração do DNS em Linux utilizando Bind9, são cobrados na Prova do LPI 201 – peso2.

9.3.2. Resolução Iterativa

Enquanto o servidor cache do exemplo acima executa um processo recursivo de consultas sucessivas descendo a árvore até a autoridade capaz de responder definitivamente ao questionamento apresentado, os servidores ".", "br.", "com.br.", apenas informam que conhecem alguém mais preciso que eles. Essa é uma consulta iterativa. Iteração, nesse caso, significa "apontar para o mais próximo conhecido".

9.4. Arquivo /etc/hosts

Derivado do arquivo HOSTS.TXT original, aquele que era atualizado e distribuído antes do surgimento do Domain Name System, o arquivo /etc/hosts continua tendo um papel muito importante no processo de resolução. No passo a passo descrito anteriormente, observe que ele é a primeira fonte de consulta do resolver.

Este comportamento pode ser modificado através do arquivo /etc/nsswitch, porém, isso só seria feito em um cenário muito particular. Podemos considerar que quase a totalidade dos sistemas *nix vão seguir a ordem de resolução padrão.

Sendo assim, é bom conhecer a sintaxe desse arquivo:

```
Endereco_IP hostname_canonico [aliases...]
192.168.200.254    gateway.com.br gateway
```

São possíveis um endereço IP por linha, seguido de um endereço canônico e opcionalmente aliases. O nome de host canônico pode ser qualquer nome no formato DNS, porém, em alguns casos, como o de servidores de e-mails, é adequado que os IPs presentes nesta máquina tenham um FQDN pertinente. Full-Qualified Domain Name é um hostname que identifica sem ambigüidade a posição daquele nó dentro da árvore DNS.

Por essa razão, um FQDN deve terminar com um ".", que representa a raiz da árvore. O aliases podem ser outros hostnames canônicos, ou simples sufixos, para simplificar a escrita de determinados endereços. Após a instalação do sistema, o arquivo hosts costuma conter uma entrada referente ao IP da interface loopback, e uma entrada para cada outra interface presente na máquina para qual um endereço foi atribuído. Por exemplo:

```
127.0.0.1 localhost
192.168.1.23 micro23.microx.com.br. micro23
```

Pode ser acrescentadas quantas entradas forem necessárias, inclusive para IPs externos. Atualmente, os sistemas baseados em Debian já contém entradas para endereçamento IPv6, que seguem a mesma lógica.

9.5. Ferramentas de consulta

Caso ainda não estejam presentes, vamos baixar instalar as ferramentas de pesquisa.

```
# aptitude install dnsutils
```

- **nslookup**

A ISC diz literalmente, no manual de utilização do BIND: "Devido a sua interface misteriosa e freqüente comportamento inconsistente, nós não recomendamos o uso do nslookup. Usem o dig no lugar dele". Porém, o pacote é mantido pela própria ISC em nome da legião de administradores que se habituaram a utilizar o nslookup como ferramenta de solução de problemas.

Dentre suas vantagens está o fato de ter uma biblioteca de resolução independente do sistema (resolvedor), e consultar um servidor por vez, dentre os listados no resolv.conf. Pode deixar a consulta mais lenta, mas torna a triagem mais controlável.Dentre os problemas mais crônicos do nslookup estão: respostas confusas e erros indefinidos.

- **host**

O comando host é concebido para dar respostas objetivas, limitando-se na maioria dos casos a uma só linha. Porém, repostas mais detalhadas podem ser obtidas com a utilização de parâmetros.

Ao contrário do dig, o host consulta a search list do arquivo /etc/resolv.conf

- **dig**

O comando dig é o acrônimo para domain information groper, que significa algo como "aquele que busca por informações de domínio no escuro", e ao mesmo tempo, a palavra dig em inglês significa literalmente "escavar". Acho que mencionar estas curiosidades demonstra o esforço de imaginação dos criadores do dig, e não à toa, ele é o comando de pesquisa mais poderoso no pacote de utilitários BIND.

No dig há dezenas de opções e incontáveis combinações entre elas, por isso consultar o man, e sobretudo, ter um forte domínio do funcionamento do sistema de nomes de domínio é necessário para dominá-las.

O dig não utiliza a opção search do /etc/resolv.conf, por isso é necessário utilizar FQDN em todas as buscas.

9.6. Prática dirigida

Para perceber as vantagens e desvantagens dos comandos a seguir, execute-os juntamente com o professor e ouça as suas explicações.

1) **Vamos começar por um nslookup interativo:**

```
# nslookup
```

2) **Exibindo as configurações do nslookup:**

```
> server
> set all
```

3) **Buscando por registros de endereços IP, entrega de e-mails, autoridade sobre o domínio e dados adicionais:**

```
> www.uol.com.br.
> set type=MX
> gmail.com.
```

4) **Exibir as informações do registro do domínio e SPF (item que eremos estudar a frente):**

```
> set type=SOA
> uol.com.br.
> set type=TXT
> terra.com.br.
```

5) **Agora vamos experimentar o objetivo comando host:**

```
# host www

# host -v www.4linux.com.br.

# host -v -t mx 4linux.com.br
# host -v -t soa 4linux.com.br

# host -l -v -t any 4linux.com.br
```

6) **E finalmente, o "verborrágico" comando dig:**

```
# dig www

# dig www.4linux.com.br.

# dig @200.204.0.138 www.4linux.com.br.

# dig -t mx 4linux.com.br.

# dig -t soa 4linux.com.br.

# dig @192.168.0.254 exemplo.com.br axfr

# dig -x 200.212.122.137

# dig +trace www.4linux.com.br.
```

9.7. BIND9

O BIND (Berkeley Internet Name Domain) é o servidor de nomes utilizado na grande maioria dos servidores da Internet, provendo uma estável e robusta arquitetura sobre a qual as organizações podem construir sua estrutura de nomes.

1) Para instalar o BIND9 no Debian basta executar:

```
# aptitude install bind9
```

O arquivo de configuração do BIND9 chama-se named.conf, e nas distribuições Red Hat e Suse ele fica exatamente no diretório /etc. No Debian, entretanto, este arquivo foi fragmentado em três. O arquivo principal ainda chama-se named.conf mas contém apenas configurações estáticas. Ele utiliza a cláusula include para anexar os arquivos named.conf.options e named.conf.local. Sendo que desses dois, o primeiro serve para personalizar todas opções referentes ao funcionamento do próprio BIND, enquanto que o segundo serve para declarar todas as zonas pelas quais este servidor deve responder.

```
# ls -lh /etc/bind
```

O arquivo db.root (/var/named/named.ca no RedHat) relaciona os endereços dos 13 servidores raiz, e é lido como zona hint, que será explicada adiante.

O BIND vai utilizar a porta 53/UDP para receber consultas, a porta 53/TCP para transferir zonas para servidores escravos, a porta 953/TCP para receber comandos via rndc (que dependem de chaves criptografadas), e portas udp altas podem ser dinamicamente atribuídas para efetuar consultas em outros servidores.

2) Vamos observar a recursividade e as portas envolvidas utilizando o tcpdump, mas antes vamos verificar as portas:

```
# netstat -nltup
```

3) Em um segundo terminal:

```
# aptitude install tcpdump
```

4) Execute o sniffer tcpdump para verificar os pacotes saindo de uma porta alta até a porta 53/udp:

```
# tcpdump -i eth0 -n port 53
```

5) No terminal anterior:

```
# dig @localhost -t any wikipedia.com
```

6) Os logs do serviço BIND serão lançados, por padrão, no arquivo /var/log/daemon.log:

```
# tail /var/log/daemon.log
```

No Red Hat e Suse os eventos do serviço serão logados diretamente no /var/log/messages.

9.8. Servidor cache

As bibliotecas do resolvedor da maioria dos sistemas operacionais não são capazes de executar o processo de resolução completo, chamado recursivo, como vimos acima. Ao invés disso elas dependem de um servidor intermediário com essa capacidade.

Quando nos conectamos de casa diretamente à Internet, o serviço DHCP do provedor se encarrega de nos atribuir o endereço de seus servidores cache. Caso contrário, nosso resolver não teria a quem consultar e não conseguiríamos navegar.

No entanto, muitos administradores de rede utilizam os IPs desses provedores para configurar várias estações de trabalho de uma rede. O efeito disto é que cada estação vai fazer suas próprias consultas individuais, multiplicando o volume de

dados trafegados através do link de Internet, desperdiçando tempo e ocupando largura de banda.

Quanto maior a rede, pior o impacto. A alternativa para evitar estes problemas é manter um servidor DNS caching only. Servidores cache reservam o resultado de suas buscas na memória pelo tempo limite estabelecido pela autoridade sobre o domínio consultado. Dessa forma, independente da quantidade de máquinas da rede, as consultas serão feitas na Internet apenas uma vez a cada intervalo de atualização.

1) **Nosso servidor BIND recém instalado já está operando como servidor cache. Para testar:**

```
# dig @localhost -t soa fsf.org
```

2) **Observe os dados no rodapé da consulta e repita o comando:**

```
# dig @localhost -t soa fsf.org
```

Observe o query time no rodapé da saída do dig.

3) **Para que a partir de agora todas nossas aplicações utilizem o potencial de nosso servidor cache, edite o arquivo /etc/resolv.conf e mantenha apenas as linhas a seguir:**

```
# vi /etc/resolv.conf
```

```
nameserver 127.0.0.1
```

9.9. Restringindo consultas

Um cuidado muito importante que devemos tomar com servidores cache é limitar quem está autorizado a utilizar esse serviço. Por padrão, o nosso BIND está liberado para acesso público, ou seja, se houver uma interface conectada diretamente a Internet, qualquer outro computador no mundo pode mandar nosso servidor procurar um determinado endereço. Ficamos vulneráveis a abusos.

1) Experimente fazer uma consulta através do servidor do seu colega:

```
# dig @192.168.200.x -t txt uol.com.br
```

2) Para evitar este comportamento, edite o arquivo /etc/bind/named.conf.options e acrescente a seguinte declaração:

```
# vi /etc/bind/named.conf.options
```

```
allow-query { 127.0.0.1; 192.168.200.x; };
```

3) Para carregar as modificações no Bind execute:

```
# /etc/init.d/bind9 reload
```

Dessa forma, nosso servidor só responderá para consultas originadas localmente.

Para carregar as modificações nos arquivos confs do Bind:

 # rndc reconfig

4) Repita a experiência com o colega:

```
# dig @192.168.200.x -t txt uol.com.br
```

É possível ser ainda mais econômico em termos de banda se ao invés de executar uma busca recursiva a cada nova consulta, encaminhar a consulta para o

servidor disponibilizado pelo provedor.

5) **Isso pode ser feito acrescentando ao named.conf.options a cláusula forwarders com o ip do provedor:**

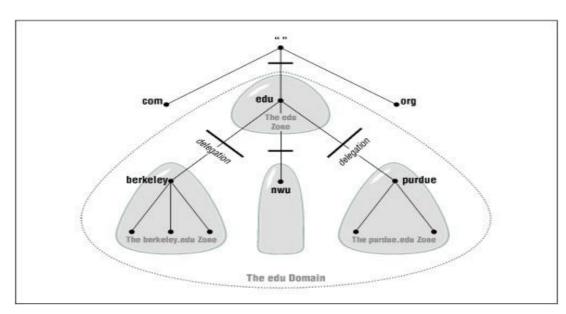
```
# vi /etc/bind/named.conf.options
```

```
forwarders {200.176.2.10; };
```

Onde IP_DO_PROVEDOR é um servidor DNS capaz de fazer consultas recursivas.

9.10. Servidor de zonas

Cada domínio na Internet tem sua autoridade, que nada mais é do que um servidor onde as informações daquele domínio são criadas, mantidas ou alteradas. Mas como um domínio pode se sub-dividir em inúmeros outros domínios, surge um outro conceito: Zonas.

Uma zona é o conjunto dos hosts de um domínio sobre o qual se mantém autoridade. Uma vez delegado um subdomínio a outra autoridade, os hosts desse domínio pertencem a zona daquela autoridade, e não mais a zona original onde inicia-se o subdomínio.

9.10.1. Tipos de zonas e Registros

Em relação as zonas, o BIND pode operar de acordo com 6 tipos: master, slave, stub, hint, forward e delegation-only.

- **master** - O BIND vai responder como autoridade sobre aquele domínio. Os dados da zona serão criados, publicados e administrados a partir dele.
- **slave** - O BIND também vai responder por esse domínio, mas nenhuma criação ou alteração respectiva a essa zona será feita localmente neste servidor. Os dados serão sempre transferidos de um servidor master.
- **stub** - Este tipo de zona não é previsto em nenhuma RFC e foi implementado apenas no bind. Assemelha-se a uma zona slave mas replica do master apenas os registros do tipo NS. Em desuso.
- **hint** - Específica para o BIND onde ele deve começar uma busca recursiva quando estiver operando como cache. Por padrão, há uma zona hint criada com os endereços dos root servers.
- **forward** - Serve para orientar o BIND a encaminhar a consulta sobre uma determinada zona para outro servidor em especial. O encaminhamento de consultas também pode ser especificado de maneira global no arquivo named.conf.options.
- **delegation-only** - Para evitar abusos de algumas autoridades sobre domínios de primeiro nível (COM, NET, ORG), o BIND mantém o tipo de zona "delegação apenas". Qualquer resposta que não tenha uma delegação explícita ou implicita na seção autoridade será transformada em uma resposta NXDOMAIN.

Vamos configurar nossa própria zona DNS que vai se chamar microx.com.br. Pode ser que na Internet já exista um domínio com este nome, mas isso não importa porque nossas consultas ficarão limitadas ao laboratório.

As zonas devem ser declaradas no arquivo named.conf.local. Uma zona mestre precisa, no mínimo do nome do domínio, tipo de zona e o caminho para o banco de dados de registros. Quando apenas o nome do arquivo é citado, o servidor BIND vai procurá-lo no diretório definido na opção directory, do arquivo named.conf.options.

Isso significa que, por padrão, o caminho completo corresponderia a /var/cache/bind/db.microx

O conteúdo do banco de dados da zona que foi declarada será principalmente uma série de registros de recursos (resources records), ou simplesmente, registros. No entanto três diretivas são suportadas:

- $TTL;
- $ORIGIN;
- $INCLUDE.

Mas com exceção do $TTL, são raramente utilizadas.

Um registro tem o seguinte formato:

dono [TTL] [classe] tipo dados.

- **dono** - É o nome do registro. Quando substituído por uma @, o dono é o próprio domínio. Caso o dono fique em branco, o BIND assume o nome do registro imediatamente superior.
- **TTL** - Um valor em segundo para a permanência dos dados deste registro no cache de um servidor. Raramente utilizado.
- **classe** - Podem ser CH, HS ou IN. O padrão é IN, de Internet, e não precisa ser declarada.
- **tipo** - No momento existem mais de 30 tipos de registro, dentre os quais veremos SOA, NS, MX, A, CNAME, TXT e PTR.
- **dados** - Diferentes tipos de dados são definidos para diferentes tipos de registros. Para um registro tipo A temos um endereço IP por exemplo.

Recentemente, registros do tipo TXT tem sido usados para aumentar o controle contra spams. São criados de acordo com o formato definido pelo projeto Sender Policy Framework, ou simplesmente SPF.

O **SPF** diz quais servidores podem ENVIAR e-mails em nome do seu domínio. O objetivo é evitar que seu domínio seja usado para forjar remetentes falsos. Grandes provedores já adotaram o SPF, e cada vez mais outros domínios vem seguindo a mesma prática. Tende a tornar-se uma imposição. Muito mais antigo que o SPF, e por consequência, uma imposição natural do ecossistema de e-mails, é garantir que o IP do **registro MX** tenha endereço reverso. Esta é uma forma de checar se o e-mail partiu de um usuário doméstico cujo computador está sendo usado como zumbi, por exemplo.

Normalmente, configurar o endereçamento reverso não depende do administrador do domínio, e sim do provedor do link. Porém, é possível requisitar autoridade sobre o bloco de IPs destinado àquele link. Vai depender do provedor. Mas como em nosso caso estamos utilizando apenas endereços privados, vamos assumir a autoridade sobre todo o bloco 192.168.200/24.

9.11. Configuração do Servidor Bind9

Agora iremos configurar o bind9, lembrando que estamos fazendo um teste interno, restringindo as consultas apenas para localhost, certifique-se dentro do arquivo /etc/resolv.conf e dentro do arquivo /etc/hosts estão corretos, efetue teste com o comando ping. Iremos prosseguir com a configuração.

1) Acrescente as linhas abaixo ao arquivo named.conf.local:

```
# vi /etc/bind/named.conf.local
```

```
zone "microx.com.br" {
    type master;
    file "db.microx";
};
zone "200.168.192.in-addr.arpa" {
    type master;
    file "rev.microx";
```

```
};
```

Agora que já declaramos as duas zonas pelas quais nos tornaremos autoridade. Vamos preencher o banco de dados de registros.

2) **Criaremos o banco de dados de registros de DNS, teremos servidores DNS, e-mail, web e ftp:**

```
# vi /var/cache/bind/db.microx
```

```
1 $TTL 1h ; TTL default para todos os registros que não tiverem seu
próprio TTL.
2 @  IN SOAns1.microx.com.br. hostmaster.microx.com.br. (
3         2009090901 ; serial
4         1h ; refresh
5         15m ; retry
6         1w ; expire
7         1h) ; negative caching TTL
8 ;
9 ;
10 @      IN   NS       ns1.microx.com.br.
11 @      IN   MX       10 mail.microx.com.br.
12 ns1    IN   A        192.168.200.X
13 mail   IN   A        192.168.200.X
14 pop    IN   CNAME    mail
15 smtp   IN   CNAME    mail
16 www    IN   A        192.168.200.X
17 ftp    IN   CNAME    www
18 @      IN   A        192.168.200.X
19 @      IN   TXT      "v=spf1 a mx ip4:192.168.200.0/24 -all"
```

Sobre o registro SOA, vão algumas explicações:

- **serial** - É a referência para os slaves saberem se a zona sofreu alterações;
- **refresh** - Tempo que o servidor secundário vai aguardar até checar se há atualizações no servidor primário;

- **retry** - Em caso de falha do refresh, o tempo até a próxima verificação;
- **expire** - O tempo que o secundário aguardará o primário voltar, se esgotar, o secundário para de responder por essa zona;
- **negative caching TTL** - Se a zona expirar, esse será o tempo pelo qual um servidor cache armazenará a informação NXDOMAIN antes de iniciar uma nova busca recursiva. O máximo são 3 horas.

Agora sobre o registro TXT:

- **a** - Qualquer registro A desse domínio;
- **mx** - O servidor de e-mail;
- **192.168.200/24** - Qualquer host da rede 192.168.200.0/24, qualquer outra origem, descarte.

3) Agora iremos configurar a zona reversa, para completar nosso registro MX:

```
# vi /var/cache/bind/rev.microx
```

```
1 $TTL 1h ; TTL default para todos os registros que não tiverem seu
próprio TTL.
2 @  IN SOAns1.microx.com.br. hostmaster.microx.com.br. (
3         2009090901 ; serial
4         1h ; refresh
5         15m ; retry
6         1w ; expire
7         1h) ; negative caching TTL
8 ;
9 ;
10 @  IN        NS        ns1.microx.com.br.
11 X  IN        PTR       mail.microx.com.br.
```

Estamos prontos para aplicar nossas configurações. Mas para observar se deu tudo certo ou não, num segundo terminal inicie o comando:

 # tail -f /var/log/daemon.log

4) Voltando ao terminal anterior:

```
# /etc/init.d/bind9 restart
```

Verifique no segundo terminal se o log não acusa nenhum problema.

5) Se o serviço levantou sem erros, teste a resolução do seu domínio:

```
# ping www.microx.com.br
# dig -t mx microx.com.br
# dig -x 192.168.200.x
# dig @localhost www.kernel.org
```

Para registrarmos um domínio público, precisamos de pelo menos dois servidores DNS respondendo pelo nosso domínio. Isso significa um servidor mestre e pelo menos um servidor escravo. A exigência é uma forma de garantir que seu domínio estará sempre disponível. Se um servidor parar, o outro continua respondendo.

Se os servidores estiverem em geograficamente separados, isso garante ainda mais disponibilidade, pois mesmo que um link caia, o outro certamente ainda estará disponível. Logo, nossa aula de BIND não estaria completa antes de configurar um servidor escravo.

Na verdade, um mesmo servidor rodando BIND pode ser simultaneamente mestre para alguns domínios, escravo para outros, e cache para todo o resto.

6) Sendo assim, cada um de nós vai se tornar escravo do colega a esquerda.

```
# vi /etc/bind/named.conf.local
```

Acrescente a zona que para qual você será escravo:

```
1 zone "nomedocolega.com.br" {
2         type slave;
3         masters { 192.168.200.x; };
4         file "sec.nomedocolega";
5 };
```

7) Ative a configuração:

```
# /etc/init.d/bind9 reload
```

8) Observe nos logs se a zona foi transferida. Em caso positivo, veja o conteúdo do arquivo criado:

```
# cat /var/cache/bind/sec.nomedocolega
```

Agora responda: Se você conseguiu baixar a zona do seu colega, quem mais conseguiria?

Para evitar abusos podemos adicionar ao arquivo /etc/bind/named.conf.options (configuração global):

9) Então para validar as ações de segurança, editamos:

```
 # vim  /etc/bind/named.conf.options
```

```
allow-transfer { none; };
```

10) Ficamos protegidos contra enxeridos, mas agora precisamos autorizar nossos escravos acrescentando:

```
# vim  /etc/bind/named.conf.options
```

Onde X é o IP do colega. No dia-a-dia do será do outro servidor.

```
allow-transfer { 192.168.200.x; 192.168.200.X;};
```

Vamos aproveitar e tornar mais rápida a atualização, enviando uma notificação sempre que fizermos uma mudança na zona, ao invés de esperar o refresh dos nossos escravos.

11) Basta acrescentar em cada zona, dentro do /etc/bind/named.con.local:

```
also-notify { 192.168.200.x; };
```

9.12. Exercícios teóricos

1) O que é DNS reverso e qual sua importância?

____

____

____

____

2) Como permitir apenas transferência de zonas para o IP 192.168.200.100?

____

____

____

3) Como permitir consultas recursivas apenas para a nossa rede?

____

____

4) O que é SPF, e qual sua relação com o Servidor DNS?

____

____

____

____

9.13. Laboratório

1. Criar cria uma view exclusiva da zona microx.com.br para consultas da rede 10.0.0.0/24;

Capítulo 10

Apache

10.1. Objetivos

- Entender a configuração básica do apache2;
- Habilitar a linguagem PHP5 no apache2;
- Configurar domínios virtuais;
- Habilitar suporte a SSL.

10.2. Introdução teórica

O Web Server Apache é um esforço comunitário feito por desenvolvedores ao redor do mundo, no qual o objetivo consiste em desenvolver um Web Server de código fonte aberto, estável e seguro. Em 1996, tornou-se um dos Web Servers mais populares no mundo, e, desde então, mantém sua posição como o servidor web com a maior base instalada no mundo.

Segundo uma pesquisa feita pelo site NetCraft, o Apache está servindo em média, 50% dos sites pesquisados. Isso só foi possível graças a uma série de qualidades, das quais algumas serão listadas na lista abaixo:

- É software livre, podendo ser modificado e adptado;
- Suporta várias linguagens, como PHP, Python, Ruby, Perl, inclusive ASP .NET;
- Multi plataforma;

- Possuí suporte a várias funcionalidades providas por módulos;
- Pode trabalhar com multi threads ou multi processos.

10.3. MPM Worker e MPM PreFork

De acordo com a documentação oficial do projeto Apachempm, é possível escolher entre algumas configurações que ajudam a otimizar a performance ou manter a compatibilidade com aplicações antigas, por exemplo. Vamos enteder as particularidades do modo PreFork e Worker.

10.3.1. MPM Pre Fork

Neste modo, o Apache trabalhará com a implementação de multi processos, de acordo com a estrutura clássica de um processo de Unix, similar a versão 1.3 do Web server em questão.

Assim sendo, um único processo será responsável por executar novos processos que serão utilizados para aguardar novas conexões e responder as requisições existentes. Este modo é ideal para quem precisa manter compatibilidade com aplicações e bibliotecas que não suportam o o modo thread.

10.3.2. MPM Worker

No modo MPM Worker, o Apache trabalhará com uma implementação mista de processos e threads, o que possibilita atenteder mais conexões simultaneas com um custo menor de hardware, já que threads, por definição, são mais velozes que processos.

Neste modo, o apache mantém uma série de threads ociosas, fazendo com que

novas conexões sejam processadas e respondidas de uma maneira mais rápida do que no modo Pre Fork. Infelizmente, nem toda aplicação se dá bem com threads, como o PHP5, por exemplo.

10.4. Prática dirigida

10.4.1. Instalação do Apache 2

1) Vamos instalar o Apache 2:

```
# aptitude install apache2
```

O Apache2 no Debian é dividido em uma série de arquivos e diretórios. Vamos conhecer estes arquivos e suas respectivas funções:

/etc/apache2/apache2.conf - Arquivo de configuração principal;
/etc/apache2/modules.conf - Arquivo de configuração de módulos;
/etc/apache2/ports.conf- Arquivo de configuração de portas;
/etc/apache2/sites-available - Configuração de sites disponíveis;
/etc/apache2/mods-available - Modulos habilitados;

2) Abra o arquivo de configurção para que possamos visualizar as principais opções de configuração do Apache:

```
#vim /etc/apache2/apache2.conf
```

A variável ServerRoot define aonde o Apache deve procurar por seus arquivos de configuração. No exemplo abaixo, os arquivos de configuração serão procurados em /etc/apache2:

```
ServerRoot /etc/apache2
```

Já a variavel DocumentRoot diz ao Apache aonde procurar os sites que deve

ser apresentados aos usuários. No exmeplo, os sites serão armazenados em /var/www:

```
DocumentRoot /var/www
```

O usuário e grupo que executa o apache são definidos pelas variáveis abaixo:

```
User $APACHE_RUN_USERS
Group $APACHE_RUN_GROUP
```

Caso seja encontrado algum erro durante o funcionamento do Apache, o mesmo será registrado de acordo com o desiginos da variável ErrorLog, como no exemplo abaixo:

```
ErrorLog /var/log/apache2/error.log
```

E, ainda falando em logs, o Apache suporta a declaração de registros personlizados através do uso da variável LogFormat:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%Refereri\" \"%User-Agenti\""
combined
```

3) Os logs, por padrão, serão armazenados no formato "combined", dentro do diretório /var/log/apache2/access.log, e o formato "combined" fará o registro dos seguintes itens:

- **%h** - Hostname ou endereço IP do visitante;
- **%l** - Hífen;
- **%u** - Nome de usuário, caso exista;
- **%t** - Horário de acesso;
- **%r** - Requisição solicitada;
- **%>s** - O resultado da solicitação ;
- **%b** - O tamanho em bytes da resposta;
- **Refereri** - O site anterior a visita, se informado;

- **User-Agenti** - O navegador do visitante, se informado.

A prova pode perguntar sobre as variaveis ErrorLog, CustomLog, ServerRoot e DocumentRoot.

10.5. Ajustes do módulo Worker e PreFork

Ainda dentro do arquivo de configuração do apache, existem ajustes de performance para os módulos MPM Worker e MPM Prefork. Por padrão, o apache vem configurado para trabalhar em MPM Worker, então, vamos entender suas configurações:

```
1 StartServers 2
2 MaxClients 150
3 MinSpareThreads25
4 MaxSpareThreads75
5 ThreadsPerChild25
6 MaxRequestsPerChild0
```

- A variavel **StartServers** configura o número inicial de servidores;
- A variável **MaxClients**, o número máximo de conexões simultaneas;
- A variável **MinSpareThreads**, configura o valor mínimo de threads em espera;
- A variável **MaxSpareThreads**, configura o valor máximo de threads em espera;
- E a varável **MaxRequestPerChild**, configura o valor máximo por processo.

O módulo PreFork também possui ajustes similares ao módulo Worker:

```
1 StartServers 5
2 MinSpareServers 5
3 MaxSpareServers10
4 MaxClients 150
5 MaxRequestsPerChild0
```

- A variavel **StartServers** configura o número inicial de servidores;
- A variável **MinSpareServer**, configura o valor mínimo de processos em espera;
- A variável **MaxSpareThreads**, configura o valor máximo de processo em espera;
- A variável **MaxClients**, o número máximo de conexões simultaneas;
- E a varável **MaxRequestPerChild**, configura o valor máximo por processo.

10.6. Segurança

Dentro do arquivo security, vamos colocar as regras de segurança:

```
# cd /etc/apache2
```

```
# vim conf.d/security
```

As variáveis abaixo ajudam a dificultar o processo de descoberta da versão do servidor e sistema operacional:

```
ServerSignature Off
ServerTokens Prod
TraceEnable Off
```

Após efetuar os ajustes, se necessário, reinicie o servidor apache:

```
invoke-rc.d apache2 restart
```

10.7. Suporte a PHP

A linguagem de programação PHP é uma das mais populares entre os desenvolvedores Web. Muitas ferramentas de blogs, WebSites e também ferramentas web para administração de serviços e servidores, como o PhpLdapAdmin, necessitam ter o PHP5 configurado como pré requisito para instalação.

1) **Instalando o suporte a PHP5:**

```
# aptitude install php5 libapache2-mod-php5
```

2) **Verifique se os módulos do PHP5 estão ativados:**

```
# ls -l /etc/apache2/mods-enabled
# apache2ctl -M
```

3) **O PHP5 é ativado por padrão durante a instalação do módulo, porém, caso necessário, habilite o suporte manualmente:**

```
#a2enmod php5
```

Na plataforma Red Hat, a habilitação de módulos é feita utilizando o comando system-config-http.

4) Reinicie o Apache:

```
# /etc/init.d/apache2 restart
```

5) Testando o PHP, para testar, crie um arquivo com o nome index.php no diretório /var/www/ com o conteúdo a seguir e abra no seu browser:

```
# vim /var/www/index.php
```

```
<?
phpinfo()
?>
```

Agora, acesse sua página WEB e verifique o resultado.

O arquivo /etc/php5/apache2/php.ini, armazena as configurações do funcionamento do php5 com apache2.
A função do arquivo e seu path completo podem ser cobrados na LPI.
O Apache2 e sua configuração prova 202 - peso 3.

10.8. Domínios virtuais

Um Domínio Virtual é uma funcionalidade que permite ao seu servido Web responder com um ou mais sites em um mesmo IP, o que possibilita acessar serviços e páginas diferentes em um mesmo servidor, apenas apontando a entrada DNS correta nos arquivos de configuração.

Os domínios virtuais devem ser configurados neste dois diretórios:

- **/etc/apache2/sites-available**: Neste diretório ficam todos os arquivos de configuração dos domínios virtuais;

- **/etc/apache2/sites-enabled**: Neste diretório ficam todos os domínios virtuais ativos, que na verdade são links simbólicos para os arquivos de configuração localizados no diretório citado anteriormente.

1) Vamos criar um domínio virtual:

```
# vim /etc/apache2/sites-available/www.microx.com.br
```

```
1  # Host Virtual
2
3  NameVirtualHostwww.microx.com.br
4
5  <VirtualHost www.microx.com.br>
6    DocumentRoot /var/www/microx.com.br
7    ServerName microx.com.br
8    ServerAdmin webmaster@microx.com.br
9    ErrorLog /var/log/apache2/microx.com.br-error.log
10   CustomLog /var/log/apache2/microx.com.br-access.log common
11  </VirtualHost>
```

A prova de certificação pode cobrar o uso de parametros como ErrorLog e sua função

2) Crie o diretório onde vai ficar hospedado o domínio virtual:

```
# mkdir /var/www/microx.com.br
```

3) Dentro do diretório recém criado, crie um arquivo chamado index.html:

```
#vim /var/www/microx.com.br/index.html
```

```
1  <html>
```

```
2 <title> Minha página <title>
3 <body>
4 Funciona! :)
5 </body>
6 </html>
```

4) Para testar a sintaxe de seu arquivo de Virtual Host:

```
# apache2ctl -S
```

5) Utilize o comando a2ensite para habilitar o domínio virtual recém criado:

```
 # a2ensite
```

6) O apache possui um comando para habilitar o domínio virtual sem a necessidade de criar os links virtuais via linha de comando:

```
# a2enmod www.microx.com.br
```

7) Recarregue as configurações sem reiniciar o apache:

```
# invoke-rc.d apache2 reload
```

Agora, no seu navegador Web, acesse o site www.microx.com.br e veja se o mesmo está funcionando.

10.9. Suporte a HTTPS

O SSL, ou Secure Sockets Layer, é um padrão Web que permite trafegar dados seníveis e confidenciais, com segurança através da internet, utilizando o protocolo HTTPS. Sendo assim, é extremamente importante que o administrador de redes configure o acesso SSL para sites que necessitam deste tipo de proteção.

1) Verifique se o módulo ssl está habilitado, e, em caso negativo, habilite-o:

```
# a2enmod ssl
```

O SSL trabalha com o conceito de certificados públicos, então, devemos efetuar os procedimentos de criação do certificado que será fornecido aos clientes.

2) Entre no diretório que irá armazenar o certificado:

```
# cd /etc/ssl
```

3) Crie a chave que será usada para assinar o certificado:

```
# openssl genrsa -out microX.key 1024
```

4) Com a chave em mãos, crie o certificado (fique atento as perguntas):

```
# openssl req -new -key microX.key -out microX.csr
```

Depois de criar o certificado, você pode enviá-lo a uma unidade certificadora, que o assinará por um valor anual, ou, caso você mesmo pode assinar o certificado, lembrando que, neste caso, o cliente dirá que o certificado não foi reconhecido por uma unidade certificadora.

5) Para enviá-lo a uma unidade certificadora:

```
# openssl x509 -req -days 365 -in microX.csr -signkey microX.key -out
microX.crt
```

6) Após gerar o certificado, configure seu domínio Virtual:

```
1  NameVirtualHost *:443
2  <VirtualHost *:443>
3    DocumentRoot /var/www/microx.com.br
4    ServerName *:443
5    ServerAdmin webmaster@microx.com.br
6    ErrorLog /var/log/apache2/microx.com.br-error.log
7    CustomLog /var/log/apache2/microx.com.br-access.log common
8    SSLEngine on
9    SSLCertificateFile /etc/ssl/microX.crt
10   SSLCertificateKeyFile /etc/ssl/microX.key
11 </VirtualHost>
12
13 NameVirtualHost www.microx.com.br:80
14 <VirtualHost www.microx.com.br:80>
15   RewriteEngine On
16   Options +FollowSymlinks
17   rewriteCond %SERVER_PORT 80
18   rewriteRule ^(.*)$ https://www.microx.com.br/$1 [R,L]
19 </VirtualHost>
```

7) Ative o mod_rewrite para que, toda vez que alguém acessar seu site, seja automáticamente redirecionado para o site com HTTPS ativado:

```
#a2enmod rewrite
```

8) Reinicie o apache:

```
#invoke-rc.d apache2 stop

#invoke-rc.d apache2 start
```

9) Agora, abra o navegador e efetue o teste no endereço:

```
http://www.microx.com.br
```

10.10. Exercício teórico

1) Você precisa alterar a porta de trabalho do seu Apache para a porta 8000. Qual seria o procedimento?

___

___

___

2) Os usuários reclamam que o site está muito lento na hora de navegar no servidor. Sabendo que você tem memória sobrando na máquina e que o problema não é a rede, qual atitude tomaria de imediato?

___

___

___

3) Seu supervisor pediu que você mudasse o diretório padrão do Apache para o diretório /sites. Quais procedimentos devem ser tomados?

___

___

___

10.11. Laboratório

1. Mude a página de erro padrão 404 para uma mensagem personalizada. DICA: Use os exemplos no próprio arquivo;
2. Descubra na documentação oficial do apache o que faz o modulo Rewrite e como ele funciona;
3. Crie um domínio virtual com o seu sobrenome.

Capítulo 11

Postfix

11.1. Objetivos

- Entender como funciona um MTA;
- Entender as funções básicas de um MTA;
- Instalar e configurar o servidor Postfix.

11.2. Introdução teórica

Na década de 70, as primeiras mensagens eram enviadas pela ARPAnet, antecessora da atual Internet.

A troca de mensagens era feita em sua maioria por estudantes, pesquisadores e profissionais dos grandes centros de pesquisa, restrita a poucos usuários que tinham acesso a essa rede. As mensagens eram enviadas através de um protocolo semelhante ao atual SMTP, que foi definido apenas em 1982.

O Sendmail era o servidor de correios mais utilizado na década de 90, causando amor e ódio aos administradores de sistema. Causava amor aqueles que tinham tempo de ler, estudar e compreender o seu funcionamento complexo e cheio de macros. Ódio para aqueles que precisavam apenas rotear suas mensagens e não havia necessidade de perder horas e mais horas tentando compreender seu funcionamento. A sua forma monolítica também era um grande ponto negativo. Sendo apenas um único processo controlando todas as etapas de transmissão de email, o Sendmail apresentava inúmeras falhas de segurança, de maior risco quando

executado em modo root. Muitos servidores eram invadidos por crackers e naturalmente os administradores de sistema procuravam alternativas.

Na época não existia muitas alternativas, os administradores continuavam a utilizar o Sendmail. Em 1998 as primeiras versões do Postfix começaram a surgir. Wietse Venema é seu criador e possui inúmeros trabalhos relacionados à segurança da informação. Wietse é pesquisador da IBM até hoje. A primeira versão oficial do Postfix, em software livre foi lançada em Dezembro de 1998.

11.3. Características do Postfix

- **Sistema multitarefa** - O Postfix possui um conjunto de módulos que desempenham um papel específico para cada etapa do tráfego de e-mails, este comportamento permite melhor desempenho em equipamentos multiprocessados.
- **Separação de privilégios** - O Postfix é executado em chroot que restringe o acesso a arquivos internos a jaula, separando assim muito de seus módulos.
- **Modular** - É possível criar módulos para trabalhar em conjunto com o Postfix, tornando-o facilmente extensível.
- **Compatibilidade** - O Postfix foi desenvolvido para suportar os formatos de armazenamentos de mensagens existentes.

Os arquivos de configuração do Postfix, podem ser encontrados no diretório /etc/postfix, onde os seus principais arquivos são:

- **main.cf** - Arquivo principal do Postfix onde ficam todas as configurações principais relacionadas ao funcionamento do Postfix.
- **master.cf** - É o arquivo que controla a ação de cada daemon do Postfix, com ele podemos dizer quantos processos smtpd estarão em execução. Caso tenhamos uma estrutura grande de máquina, uma ajuste nesses daemons serão bem compensadoras em termos de performance.

11.4. Prática dirigida

1) Vamos instalar o Postfix via aptitude:

```
# aptitude install postfix
```

No Debian, quando instalarmos o Postfix, ele automaticamente remove o servidor padrão que é o Exim4.

2) Vejamos o arquivo de configuração main.cf:

```
# vi /etc/postfix/main.cf
```

```
1 ## Banner que será mostrado nas conexões. É importante mudar.
2 smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
3 biff = no
4 ## Modificar o domínio caso o MUA's não fizer corretamente, mas
deixamos ativado,
5 ## pois isso é trabalho do próprio MUA.
6 # appending .domain is the MUA's job.
7 append_dot_mydomain = no
8
9 ## Tempo de aviso de mensagens de erro.
10 # Uncomment the next line to generate "delayed mail" warnings
11 #delay_warning_time = 4h
12
13 ## Parâmetros de criptografia.
14 # TLS parameters
15 smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
16 smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
17 smtpd_use_tls=yes
18 smtpd_tls_session_cache_database = btree:$
{queue_directory}/smtpd_scache
19 smtp_tls_session_cache_database = btree:$
{queue_directory}/smtp_scache
```

```
20
21 # See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package
for
22 # information on enabling SSL in the smtp client.
23
24 ## Nessa opção, precisamos colocar o hostname da máquina e o domínio
que é
25 ## conhecido como FQDN.
26 myhostname = mail.4linux.com.br
27
28 ## Arquivos onde são configurados os alias de e-mails.
29 alias_maps = hash:/etc/aliases
30 alias_database = hash:/etc/aliases
31
32 ## Define a origem local, que por padrão é o mesmo FQDN que está
em /etc/mailname.
33 myorigin = /etc/mailname
34
35 ## Domínios que o seu servidor pode receber mensagens.
36 mydestination = mail.4linux.com.br, localhost.4linux.com.br, ,
localhost
37
38 ## Essa opção só é usada se o seu servidor faz Relay para outros
servidores
39 ## de e-mail.
40 relayhost =
41
42 ## Nesse campo deveremos colocar apenas os IP's que podem realmente
fazer relay
43 ## em seu servidor.
44 ## CUIDADO, se adicionarmos IP's ou classes demais, o servidor poderá
virar alvo
45 ## de spammers.
46 mynetworks = 127.0.0.0/8 192.168.200.0/24
47
48 ## Padrão de entrega das mensagens. Nesse caso é usado o mbox.
49 mailbox_command = procmail -a "$EXTENSION"
50
51 ## Tamanho máximo de caixa-postal para entrega local
52 mailbox_size_limit = 0
53
54 ## Em alguns clientes, podemos adicionar um sinal espacial ao
```

```
endereço de e-mail
55 ## para direcionar mensagens a uma determinada pasta, por exemplo.
56 recipient_delimiter = +
57
58 ## Interfaces de rede a qual o Postfix pode fazer bind, ou seja,
estabelecer
59 ## conexões. O padrão do Debian seria todas as interfaces.
60 inet_interfaces = all
61 </comandoNumerado>
```

3) Podemos agora reiniciar o Postfix:

```
# /etc/init.d/postfix restart
```

4) Veja se a porta 25 SMTP está pronta para receber conexões:

```
# netstat -nltup
```

11.5. SMTP

1) Com o comando TELNET iremos testar a conexao e enviar um email:

```
1 # telnet localhost 25
2 helo gmail.com
3 mail from:seuemaildogmail@gmail.com
4 rcpt to:aluno@microx.com.br
5 data
6 subject: Teste
7 Este é um teste de email.
8 .
9 quit
```

2) Mande uma email Local:

```
# telnet localhost 25
```

```
1 mail from:root@microx.com.br
2 rcpt to:aluno@microx.com.br
3 data
4 subject: Teste
5 Este é um teste de email.
6 .
7 quit
```

3) Verifique o log de email:

```
# tail /var/log/mail.log
```

4) Verifique o diretorio de emails:

```
# cd /var/mail
# ls -la
# cat aluno
```

5) Verifique a lista de emails do servidor:

```
# mailq
```

11.6. Courier POP3 e Courier IMAP

Um servidor de email só estaria completo se tivermos a instalação de um daemon pop3 e imap. Para isso, vamos realizar alguns testes.

1) Para que o Courier funcione vamos instalar:

```
# aptitude install courier-authdaemon courier-authlib courier-base
         courier-imap courier-pop
```

2) Após este passo, edite o arquivo:

```
# vim /etc/postfix/main.cf
```

```
home_mailbox = Maildir/
DEFAULT=$HOME/Maildir/
MAILDIR=$HOME/Maildir/
```

3) Agore visualize os arquivos e crie o smtp:

```
# cat /etc/pam.d/pop3
```

```
# cat /etc/pam.d/imap
```

```
# vim /etc/pam.d/smtp

    @include common-auth
    @include common-accont
    @include common-password
    @include common-session
```

11.6.1. Criando caixas postais:

1) Vamos criar as caixas postais:

```
# maildirmake /home/aluno/Maildir
# maildirmake /home/aluno/Maildir/.Enviadas
# maildirmake /home/aluno/Maildir/.Rascunhos
```

```
# maildirmake /home/aluno/Maildir/.lixeira
# maildirmake /home/aluno/Maildir/.Spam
```

O comando maildirmake é um comando do pacote courier, veja este site: http://www.courier-mta.org/maildirmake.html

2) Ajuste as permissões:

```
# chown aluno. /home/aluno -R
```

3) Teste do POP3 na porta 110:

```
# telnet localhost 110
```

```
user aluno
pass 123456
quit
```

4) Teste do IMAP na porta 143:

```
# telnet localhost 143
```

```
a login aluno 123456
logout
```

Se você conseguir ler as mensagens, significa que o seu servidor está pronto para receber e transmitir mensagens.

Não esqueça de publicar o registro MX no seu DNS, configurar corretamente o campo TXT, também no DNS REVERSO.

11.7. Criando alias no Postfix

Podemos criar alias para que um usuário possa receber vários e-mail's diferentes na mesma conta.

1) Edite o arquivo de alias e crie um para o seu usuário:

```
# vi /etc/aliases
```

```
usuario_de_alias: usuario_existente
```

2) Para validar essas modificações e gerar o arquivo de hash, precisamos usar o comando postalias:

```
# postalias /etc/aliases
```

3) Verifique se o arquivo aliases.db foi atualizado:

```
# stat /etc/aliases.db
```

4) Agora podemos fazer um teste via telnet, enviando o e-mail para o usuário de alias:

```
# telnet localhost 25
```

5) Se tudo deu certo, a mensagem destinada ao usuário de alias, vai ser armazenada no caixa postal do usuário real:

```
# cd /var/mail
# cat usuario
```

11.8. Exercícios teóricos

1) Quais são os dois tipos de formato de caixas postais que podemos utilizar? E qual é a diferença entre eles?

2) O protocolo POP3 é um protocolo seguro? Explique.

3) Quais são os principais arquivos de configuração do Postfix, e quais são as suas funções?

4) O que acontece se configurarmos de maneira errada a opção mynetworks dentro do arquivo main.cf?

11.9. Laboratório

1. Faça um teste via telnet no seu servidor e veja se ele responde ao comando vrfy. Use o comando vrfy após o comando helo. Exemplo: vrfy usuario. Use esse comando em usuário válido e em um usuário inválido.
2. Agora que já viu como o comando vrfy pode ser prejudicial, tente desativá-lo. Dica: man 5 postconf
3. Crie uma lista chamada todos, que quando alguém enviar um e-mail para ela, todos os usuário válidos do sistema recebam a mensagem.

Capítulo 12

Web Proxy com Squid

12.1. Objetivos

- Entender como um web proxy trabalha;
- Entender ACL's e seus tipos básicos;
- Instalar e configurar o squid;
- Configurar autenticação NCSA;
- Configurar um analisador de Log's.

12.2. Introdução Teórica

Neste capítulo, iremos observar algumas particularidades do proxy web Squid,uma solução Open Source amplamente utilizada no mercado como acelerador e filtro de conteúdo Web. Observaremos também, como utilizar algumas ACL's que poderão facilmente ser aplicadas ao que chamamos de "Cenários Comuns", aonde teremos alguns sites liberados e alguns sites bloqueados. Também iremos ver como faremos para que um usuário deva efetuar autenticação e por ultimo, veremos como gerar relatórios de navegação a partir dos logs gerados pelo squid.

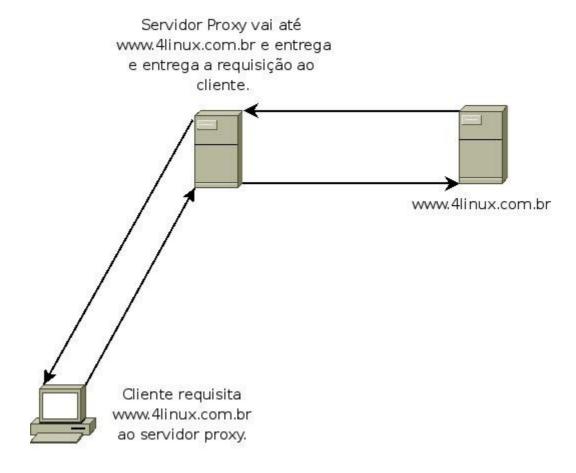
12.3. Funcionamento de um Web Proxy

Imagine que os seus usuários necessitam acessar com freqüência um site de notícias. A cada requisição, o navegador web resolve o DNS deste site, faz a requisição ao servidor web encontrado, traz o conteúdo até o usuário. Agora, imagine que 300 clientes estão acessando este site. Desperdício de banda, não acha?

As soluções Web Proxy foram desenvolvidas justamente para contornar este problema. Imagine o mesmo cenário acima, só que desta vez, ao invés de consultar o site, o navegador consulta o Web Proxy previamente configurado, e então, o servidor proxy faz a consulta ao site, só que, antes de entregar a requisição ao cliente, o servidor proxy armazena o conteúdo do site em um diretório do disco rígido e, quando um segundo cliente acessar o mesmo site, o servidor proxy verifica se o conteúdo esta armazenado em cache.

Em caso positivo, o servidor entrega o conteúdo do cache, acelerando a navegação e economizando banda. Além disso, o Web Proxy também pode agir como filtro de conteúdo, verificando se é desejável que aquele conteúdo seja acessível para aquele usuário, endereço IP ou Mac Address, e então libera ou nega o acesso de acordo com o especificado nas ACL's, item que veremos daqui a pouco. Um Web Proxy também pode ser transparente ou configurado manualmente pelo usuário ou administrador. Vamos ver as diferenças:

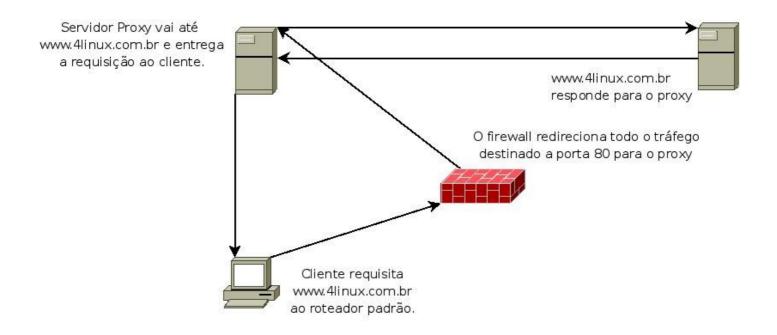
12.4. Proxy Manualmente Configurado

Em um proxy manualmente configurado, o browser sabe que é necessário fazer uma requisição ao servidor proxy, então, temos uma série de funções que o browser pode solicitar, como forçar a atualização do cache, verificar se as credenciais de autenticação foram fornecidas previamente, ou seja, o cliente sabe que deve falar com um proxy e fará a requisição direto a este. Além disso, o servidor, por sua vez, possui o numero IP do cliente que fez aquela requisição, o que possibilita criar ACL's especificas e controles de log mais apurados. A imagem a seguir mostra como funciona um proxy manualmente configurado.

12.5. Proxy Transparente

Em um proxy transparente, não há necessidade de configurarmos o browser. O cliente fara sua requisição ao gateway padrão daquela rede, e então, com uma regra de firewall previamente configurada, o gateway fará o redirecionamento para o proxy, que por sua vez realizará seu trabalho.

A maior vantagem deste modelo é que não temos a necessidade de configurar os browsers manualmente, o que seria justificável em uma rede aonde não temos o poder de manipular o computador dos usuários (como um provedor de internet, por exemplo).

Em compensação, perderemos a flexibilidade dos logs e autenticação, já que o navegador web não sabe que está passando por um proxy, e também teremos que fazer NAT para acessos para sites que utilizam HTTPS, já que o squid não sabe lidar com este tipo de conteúdo quando esta trabalhando de forma transparente.

Abaixo, uma ilustração simplificada de como esse processo funciona:

12.6. Access Control Lists

ACL, ou Access Control List, como o próprio nome diz é uma maneira de criar listas de acesso no Squid. Com elas podemos criar uma regra dizendo que a ACL de nome "MyNetork" engloba todos os endereços originados em 192.168.200.0/24, ou seja, é uma ACL que "casa" com a rede inteira. Também podemos definir uma ACL chamada "Site4linux" que "casaria" com qualquer domínio destino no qual constasse .4linux.com.br.

ACL'S de origem e destino podem ser cobrados na LPI-II, bem como sua forma de utilização.

12.7. Tipos comuns de ACL's

Basicamente, as ACL's disponíveis no squid para utilização na maioria dos casos podem ser agrupadas na seguinte lista:

- ACL's de origem
- ACL's de destino
- ACL's de horário

12.7.1. Sintaxe das ACLS

```
acl <nome da acl> <tipo da acl> <padrão da acl>
```

Vamos comentar estes três tipos nas seções seguintes:

12.8. ACL's de origem

ACL's de origem são utilizadas para controlar todo e qualquer acesso que tenha como origem um determinado padrão.

Essa origem, geralmente é um endereço de host ou endereço de rede. Também pode ser configurado um domínio, mas tenha em mente que, neste caso, é necessário que seu servidor proxy esteja completamente hábil a resolver estes endereços.

Algumas acl's de origem:

```
acl MyNetworksrc 192.168.200.0/24
acl gateway src 192.168.200.254
```

12.9. ACL's de destino

ACL's de destino são mais comuns e frequentemente utilizadas para criar padrões que casem com determinados endereços de rede, endereços de domínio, partes de um domínio e também por expressões regulares. Vamos dar uma olhada na construção dessas ACL's.

Algumas acl's de destino

```
acl Servers dst 192.168.200.1 192.168.200.2 192.168.200.3
acl 4linuxdstdomain .4linux.com.br
acl Brasildstdomain .com.br
```

12.10. ACL's de horário

ACL's de horário são muito uteis, quando queremos, por exemplo, permitir acesso a sites de relacionamento durante o horário de almoço ou fora do horário de expediente. Vamos ver uma ACL's que poderíamos utilizar para especificar o horário de almoço, e outra para especificar o horário da manhã.

```
acl almoco     time 13:00-14:00
acl parte_manha time 08:00-12:59
```

Esses são os tipos básicos de ACL's, porém, uma ACL sozinha não faz absolutamente nada. Durante a prática dirigida, veremos como tornar as ACL's úteis com configurações que permitem utilizar as ACL's para bloqueio e liberação de sites, domínios e horários.

12.11. Filtros

Outro recurso interessante para o uso de um proxy é oconceito de filtro de conteúdo, que possibilita a filtragem do conteúdo web dos usuários. Para realizar essa configuração, podemos usar os seguintes filtros:

- **src** – Filtro por rede ou endereço IP;
- **time** – Filtro por hora e dia da semana;
- **urlpath_regex** – filtro de complemento de uma url;
- **url_regex** – Filtro de uma string na url;

- **dstdomain** – Filtro de uma url;
- **proxy_auth** – Filtro por usuários autenticados;
- **arp** - Filtro por MAC address;
- **maxconn** – Filtro por conexões;
- **proto** – Fitro por protocolos;
- **port** – Filtro por porta.

12.12. Prática Dirigida

1) O primeiro passo é instalar e confirmar a instalação do Squid:

```
# aptitude install squid
# dpkg -l squid
```

Os arquivos e diretórios principais que ele utilizará:

/etc/squid/squid.conf - Arquivo de configuração

/var/log/squid/- Arquivos de log do squid.*

/var/spool/squid - Diretório que contém o cache do squid

2) Após confirmar a instalação do squid, verifique o conteúdo dos diretórios mencionados.

```
# ls /var/spool/squid
# ls /var/log/squid
# ls /etc/squid
```

3) **O squid é um software bem documentado, e boa parte desta documentação está em forma de comentários no arquivo de configuração. Vamos dar uma olhada nele.**

```
# vim /etc/squid/squid.conf
```

12.13. Configurações Iniciais

A configuração padrão do squid não permite nenhum tipo de navegação, por medidas de segurança. A primeira coisa que devemos fazer é especificar qual rede o squid deve ouvir, e também devemos especificar uma ACL do tipo origem que case com o nosso endereço IP. Vamos fazer isso.

1) **Para que o squid ouça apenas uma rede, troque o valor do parâmetro "http_port 3128" para:**

```
http_port 192.168.200.X:3128
```

2) **Outra configuração importante é o parâmetro visible_hostname. Este parâmetro diz qual será o hostname que o squid irá utilizar para resolver seu endereço local e também é o endereço que será apresentado nas páginas de informação**

```
# visible_hostname proxy.microx.com.br
```

Vamos aproveitar que estamos com a mão na massa e vamos configurar alguns parâmetros relativos ao cache.

3) **Ajuste do cache em disco: Iremos especificar 512MB de cache, com 128 diretórios e 256 subdiretórios:**

```
cache_dir ufs /var/spool/squid 512 128 256
```

4) **Definindo o cache que será armazenado em memória:**

```
cache_mem 16 MB
```

5) **Agora, iremos parar o squid, verificar a sintaxe do arquivo de configuração,gerar o cache e então reiniciaremos o squid.**

```
# invoke-rc.d squid stop
# squid -k parse
# squid -z
# invoke-rc.d squid start
```

6) **Toda vez que você mudar as ACL's você deve executar os comandos:**

```
#squid -k parse
#squid -k reconfigure
```

Depois destes ajustes iniciais, estamos prontos para criar nossas ACL's e verificar o funcionamento do squid como filtro de conteúdo.

12.14. Filtrando acessos com Squid

Uma das obrigações de um Administrador de Sistemas em alguns ambientes é controlar o que deve ou não deve ser acessível na internet a partir da rede interna. Então, para que possamos entender como trabalhar com ACL's, vamos criar o seguinte cenário, onde devemos criar um conjunto de ACL's que resulte na seguinte situação, qualquer host na minha rede deve ser impedido de navegar em qualquer domínio .com, tendo como única exceção o site www.amazon.com. Como fazer isso?

O primeiro passo é procurar a seção correta para criação de ACL's dentro do arquivo de configuração do squid. Procure uma linha semelhante a esta: # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

1) Agora, depois desta linha, crie uma acl com a rede de origem, uma acl para domínios .com e uma acl para o domínio amazon.com:

```
acl MyNetwork src 192.168.200.0/24
acl International dstdomain .com
acl Amazon dstdomain .amazon.com
```

E agora, com as ACL's criadas, iremos especificar as políticas de acesso e bloqueios com o comando http_access.

2) Procure uma string chamada http_access deny all e coloque o seguinte conteúdo, antes desta regra:

```
http_access allow MyNetwork Amazon
http_access deny MyNetwork International
http_access allow MyNetwork
http_access deny all
```

3) Configure seu browser e faça o teste de navegação para os sites:

```
www.google.com
www.amazon.com
www.google.com.br
```

4) Agora, experimente inverter a ordem das regras de acesso:

```
http_access deny MyNetwork International
http_access allow MyNetwork Amazon
http_access allow MyNetwork
```

Lembre-se que a ordem das ACL's não importa, porém, o squid fará o filtro de acordo com a ordem das regras de acesso configuradas por http_access. Lembre-se disso quando fizer troubleshooting ou criar novas regras. ;)

12.15. Blacklist e Whitelist

O conceito de Blacklist e Whitelist é muito simples:

- **Blacklist** - Uma lista de palavras que eu quero negar.
- **Whitelist** - Uma lista de domínios que eu quero liberar, mas que casam com a blacklist.

1) Para exemplificar o uso de blacklists e whitelists, vamos criar duas ACL's, uma negando o qualquer url que contenha a palavra linux, e outra liberando os sites www.linux.com e www.4linux.com.br.

Não se esqueça de limpar as ACL's e os http_access anteriores.

```
acl blacklist url_regex linux
acl whitelist dstdomain www.linux.com www.4linux.com.br
```

2) Agora crie os http_access:

```
http_access deny MyNetwork blacklist !whitelist
http_access allow MyNetWork
```

3) Agora, tente acessar os seguintes sites:

```
www.4linux.com.br
www.vivaolinux.com.br
www.linux.com
www.br-linux.org
```

Para aprender mais, acesse www.squid-cache.org e leia a documentação oficial do projeto squid.

12.16. Autenticação NCSA

O Squid possui um mecanismo de autenticação que pode trabalhar de diversas maneiras, e uma delas, a NCSA, utiliza o mesmo mecanismo de autenticação do apache. Vamos configurar esse modelo de autenticação e vamos criar uma acl que exija autenticação também.

1) Vamos configurar a autenticação no Squid:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
```

1. Linha 1 configura o autenticador
2. Linha 2 o número de processos de autenticação
3. Linha 3 configura o título da caixa de autenticação
4. Linha 4 configura o tempo de validade da autenticação

2) Gerando o arquivo de senhas:

```
# aptitude install apache-utils
# htpasswd -c /etc/squid/passwd aluno 123456
```

3) Gerando as ACL's de acesso com senha:

```
acl MyNetwork src 192.168.200.0/24
acl passwd proxy_auth REQUIRED
```

4) Configurando o acesso à internet mediante senha:

```
http_access allow MyNetwork passwd
```

12.17. Auditoria de acesso com SARG

O SARG Squid Analysis Report Generator é uma ferramenta desenvolvida pelo brasileiro Pedro Lineu Orso, cujo objetivo é analisar o arquivo /var/squid/log/access.log e gerar um relatório de acesso baseado no conteúdo acessado pelos usuários.

1) Sua instalação é bem simples, bastando apenas, no Debian, executar um aptitude:

```
# aptitude install sarg
```

2) Vamos observar seu arquivo de configuração. Fique a vontade e leia alguns comentários para entende as funções do SARG.

```
# vim /etc/squid/sarg.conf
```

3) Saia do arquivo e execute o sarg:

```
# sarg
```

4) Agora, acesse o endereço localhost/squid-reports e veja o seu conteúdo

```
http://localhost/squid-reports
```

12.18. Exercícios Teóricos

1) O que fazem as opções "squid -k reconfigure" e "squid -z"

____

____

____

2) Qual é o diretório que contém os htmls informativos do squid? Como você faria para mudá-los? (dica: man squid e squid.conf)

____

____

____

____

3) Crie ACL's e regras de acesso que permitam apenas a navegação a sites governamentais, ou seja, domínios do tipo .gov e .gov.br.

____

____

____

4) Como remover a opção que mostra a versão do squid das páginas de informação?

____

12.19. Laboratório

1. Crie um conjunto de ACL's que casem com, sites de conteúdo indesejado: www.playboy.com.br, www.youtube.com e qualquer expressão que contenha as palavras "sex", "buy" ...
2. Crie as regra de acesso necessárias para que qualquer pessoa possa navegar a vontade, porém, na hora de navegar em sites de conteúdo indesejado.

Capítulo 13

OpenLDAP

13.1. Objetivos

- Implementar e configurar servidor OpenLDAP;
- Entendimento dos principais protocolos;
- Manipular base de dados, localização e edição de objetos;
- Realizar Dump da Base de dados OpenLDAP e realizar restauração;
- Integrar OpenLDAP com Squid;
- Autenticação de cliente em base do OpenLDAP;
- Implementação de ferramenta para manipulação de base OpenLDAP via browser.

13.2. Introdução teórica

A cada dia que surgem novos sistemas nas empresas afim de resolver diversos tipos de problemas, cresce a necessidade de ter um maior controle e melhores mecanismos de busca de informação. Segurança e controle de dados é imprescindível em qualquer empresa, umas das vantagens do OpenLDAP é a possibilidade de que vários sistemas possam compartilhar de base de dados de usuários e senhas de forma centralizada e integrada.

O projeto OpenLDAP é um serviço de diretório, que utiliza o protocolo LDAP,

baseado no protocolo X.500. O OpenLDAP utiliza o trafego de dados via TCP/IP, podendo ser implementado em diversas plataformas em redes IPV4 e IPV6, possibilitando autenticação, mecanismos de segurança no uso de certificados e criptografia, podendo ser configurado para restringir acesso a socket layer, ter múltipla instâncias de banco de dados, múltiplas Threads, permite replicação e configuração do serviço de acordo com a sua necessidade através de Schema.

A organização da estrutura de dados do OpenLDAP é hierárquica, sendo referenciada a forma de Árvore, com conceito de orientação de objetos.

OpenLDAP constitui-se de:

- **slapd** - Serviço OpenLDAP
- **slurpd** - Serviço para replicação e atualização OpenLDAP
- **libraries** - Bibliotecas para implementação do OpenLDAP, com utilitários e ferramentas

13.3. Prática dirigida

1) Instalação do OpenLDAP, verifique se já tem instalado o OpenLDAP:

```
# dpkg -l slapd
```

2) Instale pacotes do OpenLDAP:

```
# aptitude install libldap2 ldap-utils slapd
```

3) Configure as opções do OpenLDAP:

```
# dpkg-reconfigure slapd
```

4) Configure quando for solicitado em:

- Omitir configuração do servidor OpenLDAP: NÃO
- Informe o nome de domínio DNS para construir a base dn: microx.com.br
- Informe nome da organização: 4linux
- Digite senha: 123456
- Escolha base de dados: BDB
- Remoção da base de dados quando o pacote slapd for expurgado: NÃO
- Mover base antiga de dados em /var/lib/ldap: SIM
- Permitir protocolo LDAPv2: SIM (Requirido para integrar o Squid com OpenLDAP)

5) Inicie o serviço do OpenLDAP:

```
# /etc/init.d/slapd start
```

6) Verifique se o serviço está disponível para a rede:

```
# netstat -nltup
```

7) Visualize a base física de dados do OpenLDAP:

```
# ls /var/lib/ldap/
```

8) Visualize o arquivo de configuração OpenLDAP no Debian:

```
# vi /etc/ldap/slapd.conf
```

```
1 # Este é o principal arquivo de configuração do slapd. Veja man page
slapd.conf
2 # para verificar demais configurações
3
4 ########################################################################
##
5
6 # Configurações Globais
7
8 # Autorizar LDAPv2 binds
9 allow bind_v2
10
11 # Definições de Schema e objectClass
12 include/etc/ldap/schema/core.schema
13 include/etc/ldap/schema/cosine.schema
14 include/etc/ldap/schema/nis.schema
15 include/etc/ldap/schema/inetorgperson.schema
16
17 # Local do arquivo onde encontra o número do processo slapd.
18 O script init.d não parará o servidor se você mudar isto.
19 pidfile/var/run/slapd/slapd.pid
20
21 # Lista de argumentos a serem passados para o servidor
22 argsfile /var/run/slapd/slapd.args
23
24 # Nível de logs a serem gerados pelo servidor, leia slapd.conf para
maiores
25 # informações
26 loglevel 0
27
28 # Local onde módulos dinâmicos são armazenados
29 modulepath  /usr/lib/ldap
30 moduleload  back_bdb
31
32 #Verificação de schema
```

```
33 schemacheck on
34
35 # O número máximo de entradas que estão retornadas para operação de
busca
36 sizelimit 500
37
38 # Parâmetros para configuração de Threads / CPU
39 tool-threads 1
40
41 ########################################################################
##
42 # Especificação de Diretivas para o bdb
43 # 'backend' directive occurs
44 backend     bdb
45 checkpoint 512 30
46
47 ########################################################################
##
48 # Especificação de Diretivas para outro:
49 # 'backend' directive occurs
50 #backend         <other>
51
52 ########################################################################
##
53 # Especificação de diretivas para banco de dados #1, tipo bdb:
54 # 'database' directive occurs
55 database bdb
56
57 # Definição de sufixo da base
58 suffix "dc=teste,dc=seu-nome,dc=br"
59
60 # rootdn diretiva para especificar um super usuário no banco de
dados.
61 # Isto é preciso.
62 # for syncrepl.
63 # rootdn "cn=admin,dc=teste,dc=seu-nome,dc=br"
64
65 # Local onde os arquivos de banco de dados são armazenados
fisicamente #1
66 directory "/var/lib/ldap"
67
68 # Para pacote Debian, não usamos 2MB como default mas tenha certeza
```

```
de atualizar
69 # este valor
70 dbconfig set_cachesize 0 2097152 0
71
72 # Verificar informações de bugs encontrados nestes parâmetros
73 # http://bugs.debian.org/303057
74
75 # Números de objetos que podem ser travados no mesmo tempo
76 dbconfig set_lk_max_objects 1500
77 #Números de lockers (ambas requisições e concessões)
78 dbconfig set_lk_max_locks 1500
79 #Números de lockers
80 dbconfig set_lk_max_lockers 1500
81
82 # Opções de indexação #1
83 index objectClass eq
84
85 # Salva o tempo que entradas foram modificadas no banco de dados
86 lastmodon
87
88 # Local onde são replicados os logs do banco de dados
89 # replogfile /var/lib/ldap/replog
90
91 # Parâmetros de acesso e permissões
92 access to attrs=userPassword,shadowLastChange
93  by dn="cn=admin,dc=teste,dc=seu-nome,dc=br" write
94  by anonymous auth
95  by self write
96  by * none
97 access to dn.base="" by * read
98
99  by dn="cn=admin,dc=teste,dc=seu-nome,dc=br" write
100  by * read
101 #access to dn=".*,ou=Roaming,o=morsnet"
102 # by dn="cn=admin,dc=teste,dc=seu-nome,dc=br" write
103 # by dnattr=owner write
104 # Especificação de diretivas de banco de dados #2, de tipo outro,
pode
105 ser bdb:
106 # 'database' directive occurs
107 #database <other>
```

```
108 # Definição de sufixo da base#2
109 #suffix          "dc=debian,dc=org"
110
111 ####################################################################
###
112 </comandoNumerado>
```

9) Verificação de teste da configuração do arquivo do OpenLDAP:

```
# slaptest
```

É necessário criar a nossa base de dados. Para tal utilizaremos migrationtools, uma ferramenta para migração de base de dados escrita em Perl para o OpenLDAP. Portanto verifique e instale os pacote migrationtools e perl:

10) Verifique se tem o pacote perl instalado:

```
# dpkg -l | grep perl
```

11) Instale Perl, caso não esteja instalado:

```
# aptitude install perl
```

12) Verifique o pacote Migration Tools:

```
# aptitude search migrationtools
```

13) Realize a instalação Migration Tools:

```
# aptitude install migrationtools
```

14) Acesse o diretório /usr/share/migrationtools e copie o arquivo de configuração:

```
# cd /usr/share/migrationtools
# cp -av migrate_common.ph migrate_common.ph.original
```

15) Edite o arquivo migrate_common.ph e os campos a seguir, salve:

```
$DEFAULT_MAIL_DOMAIN="microx.com.br";
$DEFAULT_BASE=dc="microx,dc=com,dc=br";
```

16) Vamos migrar a base de usuários do sistema (/etc/passwd) para uma base padrão LDIF, para inserir na base LDAP:

```
# cd /usr/share/migrationtools
# ./migrate_passwd.pl /etc/passwd /etc/ldap/users.ldif
```

17) Verifique o arquivo /etc/ldap/users.ldif criado e observe o conteúdo:

```
# less /etc/ldap/users.ldif
```

```
1 gidNumber: 106
2 homeDirectory: /var/lib/gdm
3 gecos: Gnome Display Manager
4
5 dn: uid=root,ou=People,dc=seu-nome,dc=com,dc=br
6 uid: root
7 cn:: cm9vdA==
8 objectClass: account
9 objectClass: posixAccount
10 objectClass: top
11 objectClass: shadowAccount
12 userPassword: {crypt}$1$dL7nEggA$P6Ib/H9QBkdd/sTcUBW1z1
13 shadowLastChange: 12495
14 shadowMax: 99999
15 shadowWarning: 7
16 loginShell: /bin/bash
17 uidNumber: 0
18 gidNumber: 0
19 homeDirectory: /root
20 gecos: root
```

18) Agora vamos migrar a base de grupos do sistema (/etc/group) para uma base padrão LDIF, para inserir na base LDAP:

```
./migrate_group.pl /etc/group /etc/ldap/groups.ldif
```

19) Verifique os arquivos gerados em /etc/ldap/group.ldif:

```
# cat /etc/ldap/groups.ldif
```

20) Criando nossa base ldif:

```
# ./migrate_base.pl > /etc/ldap/base.ldif
```

21) Edite o arquivo gerado em /etc/ldap/base.ldif e remova as linhas de 1 a 10, por default durante a migração estas linhas foram criadas, podendo gerar erro durante a importação, dentro do vim execute o comando ESC:1,10d para apagar as 10 primeiras linhas:

```
# vim /etc/ldap/base.ldif
```

22) Adicione o base ldif na base do OpenLDAP:

```
# ldapadd -x -D cn=admin,dc=microx,dc=com,dc=br -f /etc/ldap/base.ldif
-W
```

23) Realize uma busca na sua base de dados OpenLDAP:

```
ldapsearch -x | more
```

24) Adicione o group.ldif:

```
# ldapadd -x -D cn=admin,dc=seu-nome,dc=com,dc=br -f
/etc/ldap/groups.ldif -W
```

25) Adicione o user.ldif:

```
# ldapadd -x -D cn=admin,dc=seu-nome,dc=com,dc=br -f
/etc/ldap/users.ldif -W
```

26) Você pode transferir as informações em formato ldif através do slapadd passando a informação direto para servidor:

```
# /etc/init.d/slapd stop
# slapadd -l meuarquivo.ldif -f slapd.conf
```

27) Realizando busca específica através do nome do objeto que consta na base do OpenLDAP:

```
# ldapsearch -x -b 'dc=microx,dc=com,dc=br' '(cn=cdrom)'
```

28) Consulta da Base OpenLDAP:

```
# slapcat | more
```

13.4. Configurando um cliente LDAP

Para que os nossos computadores possam autenticar-se como clientes LDAP, é necessário que modifiquemos uma série de arquivos do PAM e também é necessária a instalação de alguns pacotes.

1) Instalando os pacotes necessários e configure-os:

```
#aptitude install libnss-ldap libpam-ldap
```

2) Acrescente as opções ldap nas linhas passwd, group e shadow, para que nossa estação passe a buscar os usuários no LDAP.

```
# vim /etc/nsswitch.conf
    passwd:compat ldap
    group: compat ldap
```

```
    shadow:compat ldap
```

3) Agora vamos configurar o arquivo /etc/libnss-ldap.conf:

```
# vim /etc/libnss-ldap.conf
```

4) Procure as linhas 192 e 193 e configure-as:

```
nss_base_passwd ou=People,dc=microx,dc=com,dc=br
nss_base_shadow ou=People,dc=microx,dc=com,dc=br
```

5) Agora vamos configurar as linhas 172 e 173 do arquivo /etc/pam_ldap.conf:

```
nss_base_passwd ou=People,dc=microx,dc=com,dc=br
nss_base_shadow ou=People,dc=microx,dc=com,dc=br
```

Por ultimo, iremos configurar o PAM para que ele busque os usuários na base LDAP.

6) Configure o arquivo /etc/pam.d/common-auth :

```
1 #
2 # /etc/pam.d/common-auth - authentication settings common to all
services
3 #
4 # This file is included from other service-specific PAM config files,
5 # and should contain a list of the authentication modules that define
6 # the central authentication scheme for use on the system
7 # (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use
the
8 # traditional Unix authentication mechanisms.
9 #
10 auth sufficient pam_ldap.so
11 auth required pam_unix.so nullok_secure try_first_pass
```

7) Configure o arquivo /etc/pam.d/common-account

```
1 #
2 # /etc/pam.d/common-account - authorization settings common to all
services
3 #
4 # This file is included from other service-specific PAM config files,
5 # and should contain a list of the authorization modules that define
6 # the central access policy for use on the system. The default is to
7 # only deny service to users whose accounts are expired in
/etc/shadow.
8 #
9 account sufficient pam_ldap.so
10 account required pam_unix.so
11 session required pam_mkhomedir.so skel=/etc/skel umask=0077
```

8) Agora configure o arquivo /etc/pam.d/common-password:

```
1 #
2 # /etc/pam.d/common-password - password-related modules common to all
services
3 #
4 # This file is included from other service-specific PAM config files,
5 # and should contain a list of modules that define the services to be
6 #used to change user passwords. The default is pam_unix
7
8 # The "nullok" option allows users to change an empty password, else
9 # empty passwords are treated as locked accounts.
10 #
11 # (Add `md5' after the module name to enable MD5 passwords)
12 #
13 # The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option
in
14 # login.defs. Also the "min" and "max" options enforce the length of
the
15 # new password.
16
17 passwordsufficientpam_unix.so nullok obscure min=4 max=8 md5
18 passwordrequired pam_ldap.so try_first_pass
19
20 # Alternate strength checking for password. Note that this
```

```
21 # requires the libpam-cracklib package to be installed.
22 # You will need to comment out the password line above and
23 # uncomment the next two in order to use this.
24 # (Replaces `OBSCURE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
25 #
26 # password required    pam_cracklib.so retry=3 minlen=6 difok=3
27 # password required    pam_unix.so use_authtok nullok md5
```

9) Edite também o arquivo /etc/pam.d/common-session:

```
1 #
2 # /etc/pam.d/common-session - session-related modules common to all
services
3 #
4 # This file is included from other service-specific PAM config files,
5 # and should contain a list of modules that define tasks to be
performed
6 # at the start and end of sessions of *any* kind (both interactive
and
7 # non-interactive). The default is pam_unix.
8 #
9 session sufficient pam_ldap.so
10 session required pam_unix.so
```

13.5. Acessando o OpenLDAP via Browser com PhpLdapAdmin

Vamos acessar a base do OpenLDAP via Browser, para tal será necessário instalar php e dar suporte ao Apache.

1) Instale os pacotes do PHP necessário para o OpenLDAP:

```
# aptitude install php-pear php5-ldap
```

2) Verifique se seu servidor Apache está com suporte a PHP:

```
# ls -l /etc/apache2/mods-enable
```

3) Instalando os pacotes do phpldapadmin:

```
# aptitude install phpldapadmin
```

4) Abra o seu browser e digite no campo URL:

```
127.0.0.1/phpldapadmin/index.php
```

13.6. Autenticando o Squid na base de usuários LDAP

No nosso caso, queremos que os usuários do nosso servidor OpenLDAP sejam autenticados. Para isso, usaremos o programa ldap_auth.

1) Modifique as seguintes linhas no arquivo /etc/squid/squid.conf para ser feita a autenticação via OpenLDAP:

```
# vim /etc/squid/squid.conf
```

```
auth_param basic program /usr/lib/squid/ldap_auth -b dc=seu-
nome,dc=com,dc=br -f  uid=%s 192.168.200.X
```

Não esquecer de tirar a outra linha de autenticação usando ncsa.

13.7. Exercício teórico

1) Qual a diferença em realizar os comandos ldapsearch -x e slapcat?

____

____

____

____

____

2) O que é um schema?

____

____

____

3) Qual é o banco de dados utilizados pelo OpenLDAP e onde está armazenado no sistema?

____

____

____

13.8. Laboratório

1. Instale o LDAP tendo o seu nome como cn, como no exemplo da apostila;
2. Crie um usuário diretamente no LDAP e autentique no SQUID.

Capítulo 14

Firewall

14.1. Objetivos

- Entender as tabelas principais do iptables;
- Entender as políticas básicas;
- Entender o conceito de exceções;
- Entender o conceito de nat e forward;
- Configurar um firewall simples.

14.2. Introdução teórica

Os sistema GNU/Linux com Kernel série 2.4 e 2.6 trabalham com o Iptables para fazer o gerencialmento de regras de Firewall. Lembrando que o Iptables é apenas um Front-End que gerencia o suporte Netfilter no Kernel. O Iptables possuí 4 tabelas, sendo elas:

- **filter**
- **nat**
- **mangle**
- **raw**

A tabela filter é a tabela padrão do Iptables.

Cada uma dessas tabelas possuí o que chamamos de CHAINS. As CHAINS são onde vão ser definidos as regras para o nosso firewall.

Nesse capítulo vamos tratar mais das CHAINS da tabela filter e algumas coisas sobre as CHAINS da tabela nat.

As CHAINS da tabela filter são as seguintes:

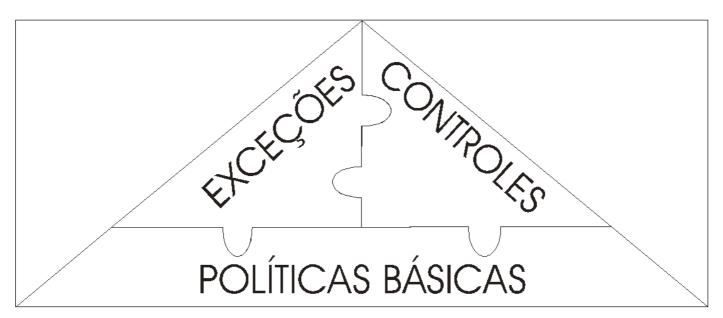
- Tabela Filter
  - **INPUT** - Regras de entrada de pacotes.
  - **OUTPUT** - Regras de saída de pacotes.
  - **FORWARD** - Regras de passagem de pacotes pelo firewall.

As CHAINS da tabela nat são as seguintes:

- Tabela Nat
  - **PREROUTING** - Regras que serão processadas antes do roteamento dos pacotes nas interfaces do firewall.
  - **POSTROUTING** - Regras que serão processadas pós roteamento dos pacotes nas interfaces do firewall.
  - **OUTPUT** - Regras de saída de pacotes.

14.3. Compreendendo as políticas BÁSICAS e o conceito das EXCEÇÕES

A metodologia utilizada para implementação do firewall será a seguinte:

Iremos negar todo o tráfego para as CHAINS de INPUT, OUTPUT e FORWARD da tabela filter, posteriormente iremos definir a relação dos serviços que devem ser liberados no firewall, a estes, iremos chamar de exceções. Todo o tráfego de pacotes que as nossas exceções não cobrir serão bloqueado por padrão. Em suma, o que não for oficialmente permitido já está expressamente negado.

14.4. Prática dirigida

Vamos agora montar o nosso Firewall.

1) **Verifique como estão configuradas as políticas básicas que estão definidas por padrão:**

```
# iptables -n -L
```

2) **Modifique as políticas básicas para DROP ALL:**

```
# iptables -P INPUT DROP
# iptables -P OUTPUT DROP
# iptables -P FORWARD DROP
```

3) Verifique se a nova política foi assumida:

```
# iptables -n -L
```

Agora que percebemos que temos um firewall ativo, devemos pensar nas demais políticas, uma vez que, por mais seguro que seja um firewall, cuja política base seja negar tudo, não é um firewall prático, pois precisamos realizar comunicações. Dessa forma, precisamos definir políticas de exceções para o Firewall.

4) Realize o teste usando o comando ping na sua interface loopback:

```
# ping 127.0.0.1
```

5) O teste anterior nos permitiu verificar que devemos definir uma política de exceção para a interface loopback. Criaremos uma política que possibilite isso:

```
# iptables -A OUTPUT -d 127.0.0.1 -j ACCEPT
# iptables -A INPUT -d 127.0.0.1 -j ACCEPT
```

6) Liste as políticas ativas:

```
# iptables -n -L
```

7) Vejamos se agora conseguimos fazer um ping na interface de loopback:

```
# ping 127.0.0.1
```

8) Execute o comando ping, tendo como alvo o endereço da máquina do instrutor para verificar se alguma comunicação é possível:

```
# ping 192.168.200.254
```

9) **Agora criaremos uma política que permita que seja executado o comando ping a partir de sua máquina com a sua interface de rede interna, mas sua máquina não irá responder a ping:**

```
# iptables -A OUTPUT -p icmp --icmp-type 8 -s 192.168.200.x  -d 0/0 -j
ACCEPT
```

```
# iptables -A INPUT -p icmp --icmp-type 0 -s 0/0   -d 192.168.200.x -j
ACCEPT
```

10) **Verifique se as regras foram adicionadas:**

```
# iptables -n -L
```

11) **Podemos ver se conseguimos fazer um ping na máquina do instrutor:**

```
# ping 192.168.200.254
```

12) **Agora que já temos um política de exceção, tente fazer um ping no domínio www.uol.com.br:**

```
# ping www.uol.com.br
```

13) **Apesar de conseguirmos usar o ping nos endereços IP's, ainda não conseguimos fazer um ping por nomes. Vamos desenvolver a regra que faça isso:**

```
# iptables -A OUTPUT -p udp -s 192.168.200.x --sport 1024:65535 -d 0/0
--dport 53 -j ACCEPT
```

```
# iptables -A INPUT -p udp -s 0/0 --sport 53 -d 192.168.200.x --dport
1024:65535 -j ACCEPT
```

14) Verifique se as regras foram adicionadas:

```
# iptables -n -L
```

15) Com as regras definidas, podemos fazer um ping por nomes:

```
# ping www.uol.com.br
```

Mesmo que liberamos o nosso firewall para resolver os nomes, ainda não conseguimos acessar um servidor Web por ele, pois precisamos liberar o acesso as portas 80 e 443.

16) Criaremos uma regra de exceção que permita navegação web:

```
# iptables -A OUTPUT -p tcp -s 192.168.200.x --sport 1024:65535 -d 0/0
--dport 80 -j ACCEPT
```

```
# iptables -A OUTPUT -p tcp -s 192.168.200.x --sport 1024:65535 -d 0/0
--dport 443 -j ACCEPT
```

```
# iptables -A INPUT -p tcp -s 0/0 --sport 80 -d 192.168.200.x --dport
1024:65535 -j ACCEPT
```

```
# iptables -A INPUT -p tcp -s 0/0 --sport 443 -d 192.168.200.x --dport
1024:65535 -j ACCEPT
```

17) Façamos um teste para ver se conseguimos traçar rotas usando a ferramenta mtr:

```
# mtr 200.176.2.11
```

18) O mtr utiliza respostas icmp do tipo 11, então precisamos criar uma regra liberando a entrada desse tipo de pacote:

```
# iptables -A INPUT -p icmp --icmp-type 11 -s 0/0 -j ACCEPT
```

19) Verifique se a regra foi adicionada:

```
# iptables -n -L
```

20) Execute o comando mtr para testar a regra criada:

```
# mtr 200.17.2.11
```

14.5. Firewall como gateway de rede

Se o nosso servidor é, por exemplo, um firewall de fronteira entre a sua rede e a internet, ou seja, um gateway de rede, devemos estabelecer uma política que faça o repasse dos pacotes de uma rede para a outra, para permitir o repasse(forward) de pacotes entre uma rede e outra.

1) A primeira coisa que precisamos fazer é liberar o repasse de pacotes entre as interfaces no kernel:

```
# sysctl -a | grep ip_forward
# sysctl -w net.ipv4.ip_forward=1
```

Para deixar esse valor fixo, devemos deixar esse parâmetro dentro de /etc/sysctl.conf

 # vim /etc/sysctl.conf

net.ipv4.ip_forward=1

2) Agora precisamos permitir no iptables que nossa rede se comunique com outras. Devemos fazer isso acrescentas regras na chain FORWARD:

```
# iptables -A FORWARD -s 192.168.200.0/24 -j ACCEPT
# iptables -A FORWARD -d 192.168.200.0/24 -j ACCEPT
```

Não podemos esquecer que a internet trabalha com IP's reservados, diferente da nossa rede. Por isso teremos que fazer a tradução do endereçamento inválido (da LAN) para o válido (da internet), através da especificação da tabela Nat, fazendo o mascaramento.

3) Vamos fazer com que nossa LAN seja mascarada:

```
# iptables -t nat -A POSTROUTING -o ethX -s 192.168.200.0/24 -j
MASQUERADE
```

A interface ethX é a que está com o IP válido.

4) Verifique como estão as regras inseridas:

```
# iptables -n -L
# iptables -n -L -t nat
```

5) Para não perdermos essas regras, podemos salvá-las utilizando recursos do iptables, lembrando que isso não é ainda um script profissional:

```
# iptables-save > /root/firewall.back
# cat firewall.back
```

6) Agora podemos fazer um teste e limpar todas as regras adicionadas na memória:

```
# iptables -F
# iptables -F -t nat
```

7) Verifique se as regras foram apagadas:

```
# iptables -n -L
# iptables -n -L -t nat
```

8) Modifique as políticas básicas para ACCEPT:

```
# iptables -P INPUT ACCEPT
# iptables -P OUTPUT ACCEPT
# iptables -P FORWARD ACCEPT
```

14.6. Script de firewall

Todas as regras que foram feitas, ficam na memória do computador, se ele for reiniciado, perderemos todas elas. Podemos utilizar o iptables-save, mas ele não fica um script profissional.

Segue aqui um script com todas as regras que foram feitas, em seguida esse script pode ser adicionado aos níveis de execução do sistema, para ser carregado sempre a máquina for ligada.

Vamos chamar nosso script de firewall.sh:

```
# cd /etc/init.d
# vim firewall.sh
```

```
1 #!/bin/bash
2 # Firewall personalizado - Curso 452 - 4Linux
3
4 ## Definição de variáveis
5
6 IPT=$(which iptables)
7 ET0="192.168.200.X"
8 NET="0/0"
9 PA=1024:65535
10 REDE="192.168.200.0/24"
11
12 ## Fechando as Políticas
13 $IPT -P INPUT DROP
14
15 $IPT -P OUTPUT DROP
16
17 $IPT -P FORWARD DROP
18
19 ## Liberando LoopBack
20
21 $IPT -A OUTPUT -d 127.0.0.1 -j ACCEPT
22
23 $IPT -A INPUT -d 127.0.0.1 -j ACCEPT
24
25 ## Liberando Ping (Saída de icmp 8 e Entrada de icmp 0)
26
27 $IPT -A OUTPUT -p icmp --icmp-type 8 -s $ET0 -d $NET -j ACCEPT
28
29 $IPT -A INPUT -p icmp --icmp-type 0 -s $NET -d $ET0 -j ACCEPT
30
31 ## Liberando resolução de nomes
32
33 $IPT -A OUTPUT -p udp -s $ET0 --sport $PA -d $NET --dport 53 -j
ACCEPT
34
35 $IPT -A INPUT -p udp -s $NET --sport 53 -d $ET0 --dport $PA -j ACCEPT
36
37 ## Liberando navegação web
38
39 $IPT -A OUTPUT -p tcp -s $ET0 --sport $PA -d $NET --dport 80 -j
ACCEPT
40 $IPT -A OUTPUT -p tcp -s $ET0 --sport $PA -d $NET --dport 443 -j
ACCEPT
```

```
41
42 $IPT -A INPUT -p tcp -s $NET --sport 80 -d $ET0 --dport $PA -j ACCEPT
43 $IPT -A INPUT -p tcp -s $NET --sport 443 -d $ET0 --dport $PA -j
ACCEPT
44
45 ## Liberando consultas mtr
46
47 $IPT -A INPUT -p icmp --icmp-type 11 -s $ET0 -j ACCEPT
48
49 ## Regras de FORWARD e NAT para liberar a LAN para acessar a
internet.
50
51 net.ipv4.ip_forward=1
52
53 $IPT -A FORWARD -s $REDE -j ACCEPT
54
55 $IPT -A FORWARD -d $REDE -j ACCEPT
56
57 $IPT -t nat -A POSTROUTING -o ethX -s $REDE -j MASQUERADE
```

Agora podemos configurar as permissões de execução para o script:

 # chmod 755 firewall.sh

 # ls -l firewall.sh

Para que ele seja iniciado junto com sistema quando a máquina for ligada, podemos colocar o script nos níveis de execução:

```
# update-rc.d firewall.sh defaults
# ls -l /etc/rc2.d
```

14.7. Exercícios teóricos

1) Quantas tabelas o iptables possuí e qual delas é a padrão?

2) Qual tabela é responsável por fazer o mascaramento de IP's inválidos?

3) Qual seriam as regras para liberar que outras máquinas consigam conectar na porta 22 do seu servidor?

4) Consulte o manual do iptables e responda qual módulo é necessário para limitar a quantidade de pacotes icmp que sua máquina pode receber, e como devo usá-lo.

14.8. Laboratório

1. Imaginem que o seu servidor tem 2 IP's, um válido e outro inválido. Crie uma regra que redirecione todos os pacotes que chegarem na porta 25 do IP válido para uma máquina da rede que tenha IP inválido na porta 25. Dica: man iptables.

Capítulo 15

OpenVPN

15.1. Objetivos

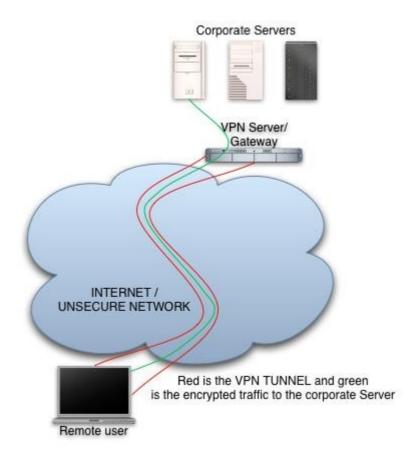
- Entender como funciona uma VPN;
- Configurar uma VPN host to host.

15.2. Introdução Teórica

VPN Virtual Private Network, é uma rede de comunicação particular, geralmente utilizando canais de comunicação inseguros, como a própria LAN ou mesmo a Internet. O que torna esta rede de comunicação particular é o fato das ferramentas de VPN empregarem métodos e protocolos de criptografia, criando um túnel de criptografia para prover acesso seguro a partes da rede ou mesmo ligação entre LAN's geograficamente separadas, eliminando a necessidade de um canal de comunicação privativo de alto custo fornecido pela operadora de telecomunicações.

Também podemos utilizar uma ferramenta de VPN para implementar ou reforçar a segurança de acesso há algum serviço dentro de nossa rede.

Você possui um software de geração de notas fiscais, e os funcionários acessam este terminal via telnet, que é um protocolo que não implementa criptografia. Para corrigir esta situação e reforçar a segurança deste ambiente, você poderia configurar uma VPN entre o computador dos usuários e o servidor, melhorando assim a segurança deste serviço.

Neste laboratório, iremos configurar uma VPN host-to-host.

15.3. Prática Dirigida

Nós trabalharemos neste laboratório com um par de chaves simétricas, ou seja, usaremos a mesma chave tanto para o servidor VPN quanto para o cliente VPN, logo, a chave deve ser gerada no servidor e replicada para o cliente via SSH.

15.3.1. Configurando o servidor

1) O primeiro passo é instalar o software OpenVPN:

```
# aptitude install openvpn
```

Vamos entrar no diretório /etc/openvpn e gerar a chave:

```
# cd /etc/openvpn
# openvpn --genkey --secret /etc/openvpn/chave
```

2) Vamos gerar o arquivo de configuração do servidor:

```
# vim /etc/openvpn/server.conf
```

```
1 #Configuração para servidor
2 dev tun
3 # Server -> 172.16.0.1
4 # Client -> 172.16.0.2
5 #Definindo os IP's da VPN
6 ifconfig 172.16.0.1 172.16.0.2
7 #Definido a chave
8 secret /etc/openvpn/chave
9 # Definindo a porta
10 port 5000
11 comp-lzo
12 verb 4
```

15.4. Configurando o cliente

1) Também é necessário ter o OpenVPN no cliente:

 # aptitude install openvpn

2) Vamos entrar no diretório /etc/openvpn e copiar a chave do servidor.

```
# cd /etc/openvpn
# scp 192.168.200.X /etc/openvpn/chave .
```

3) Vamos gerar o arquivo de configuração do cliente:

```
# vim /etc/openvpn/client.conf
```

```
1 #Configuração para cliente
2 dev tun
3 # Server -> 172.16.0.1
4 # Client -> 172.16.0.2
5 #Definindo os IP's da VPN
6 ifconfig 172.16.0.2 172.16.0.1
7 #Definindo o IP real do servidor
8 remote 192.168.200.X
9 #Definido a chave
10 secret /etc/openvpn/chave
11 # Definindo a porta
12 port 5000
13 comp-lzo
14 verb 4
```

4) Iniciando a VPN, tanto no servidor quanto no cliente.

```
# openvpn --config /etc/openvpn/server.conf
# openvpn --config /etc/openvpn/client.conf
```

5) Execute um ifconfig para ver se a interface tun0 foi criada:

ifconfig -a

6) Execute um ping na sua interface VPN:

```
# ping 172.16.0.X
```

7) Execute um ping na interface do seu colega:

```
# ping 172.16.0.X
```

Efetue testes em todos os serviços utilizando o IP da VPN.

REFERÊNCIAS BIBLIOGRÁFICAS

Nelson Mendonça e Tiago Vilas Boas. GNU Linux. Editora Brasport.

The Linux Documentation Project, *http://www.tldp.org*

Guia Foca Gnu/Linux, *http://focalinux.cipsga.org.br*

ANEXOS

System Imager - 4Linux

O que é

O System Imager é um sistema de automatização para rotinas de backup e recuperação de máquinas. Ele permite que as mesmas máquinas possam ser compartilhadas por vários cursos simultâneos, mas de forma que o estado delas, isto é, todos os seus arquivos e configurações sejam guardados e recuperados de forma individual por aluno.

Instalando o programa.

Primeiramente, o aluno deverá fazer o download do programa e mudar as suas permissões. Esse procedimento só é necessário na primeira aula.

```
 # cd /sbin
# wget 192.168.1.1/si/si_cliente
# chmod u+x /sbin/si_cliente
```

*15.4.1. **Backup ao final de cada aula***

Ao **final** de cada aula o aluno executa o comando abaixo, selecionando a opção de **Enviar Imagem.**

```
# si_cliente
```

O professor verificará se todas as maquinas estão com a imagem pronta para enviar. Em caso positivo, irá executar um programa para recebê-las.

PORTANTO: NÃO desligue seu micro, pois o servidor estará conectado a ele recebendo os arquivos modificados. Após o processo ter sido concluído, as máquinas serão desligadas automaticamente.

15.4.2. Restore antes de cada aula

No início de cada aula, a imagem de cada máquina deverá ser restaurada. Para isso, basta executar o comando abaixo, selecionando a opção Receber Imagem. Normalmente, esta operação é realizada pelo próprio instrutor antes da aula se iniciar e deverá ser realizada pelo aluno apenas sob sua orientação.

ATENÇÃO! Todos os arquivos do sistema poderão ser apagados! Se você não fez nenhum tipo de backup, faça-o antes!

```
# si_cliente
```

Aparecerá uma listagem das imagens disponíveis. Você deverá escolher aquela que corresponder a sua máquina.

Um exemplo de nomenclatura das imagens:

```
 451-Instrutor-31-Noturno-10
(Cod. do curso)-(Nome do Instrutor)-(Dia de Início)-(Período)-(Fim do IP do Micro)
```

Após a conclusão do processo, a máquina irá se reinicializar automaticamente e, em seguida, estará pronta para uso.